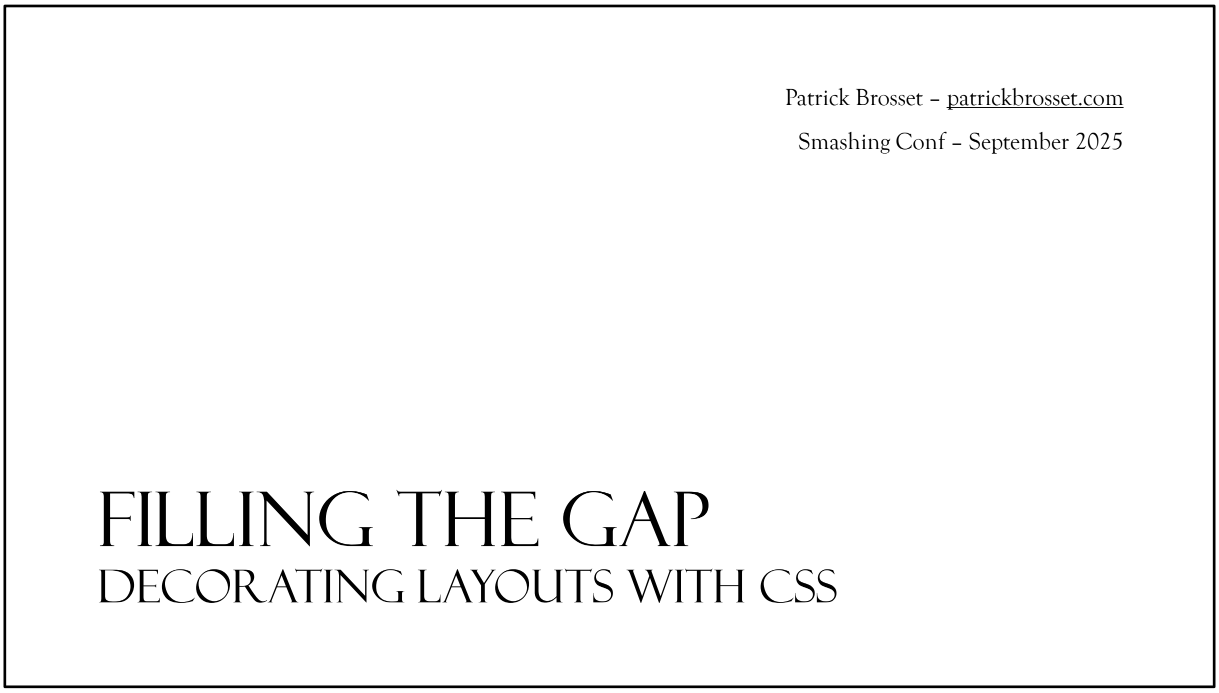


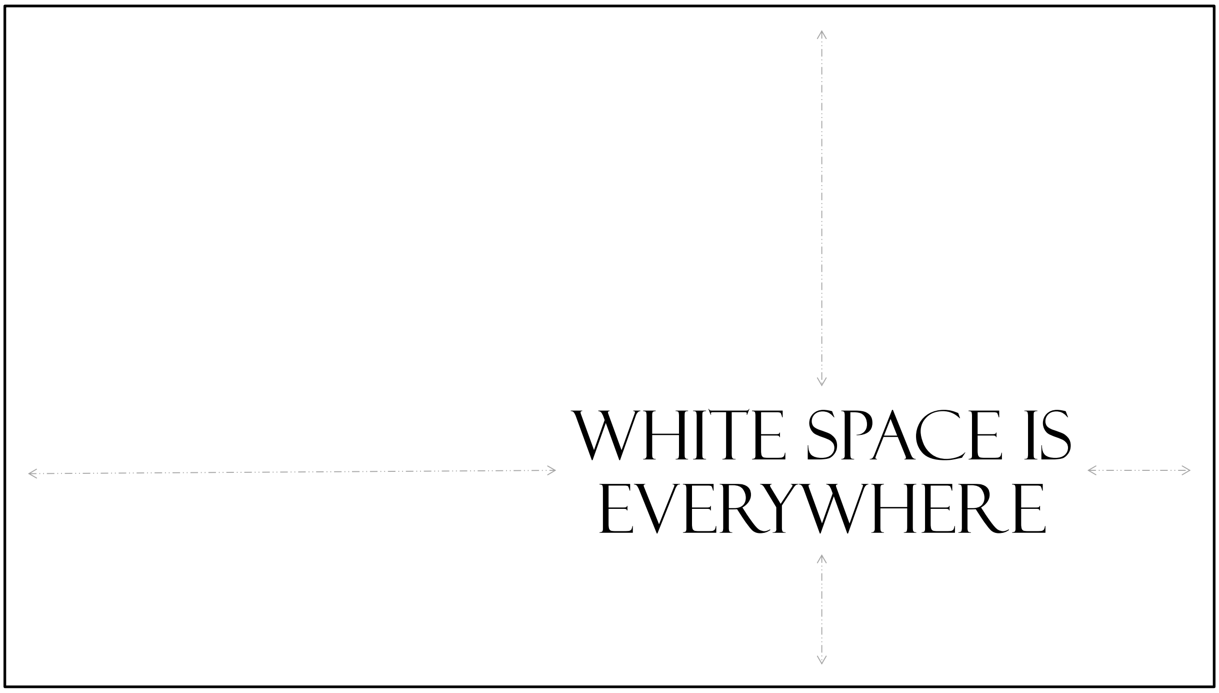
Patrick Brosset – patrickbrosset.com

Smashing Conf – September 2025

FILLING THE GAP

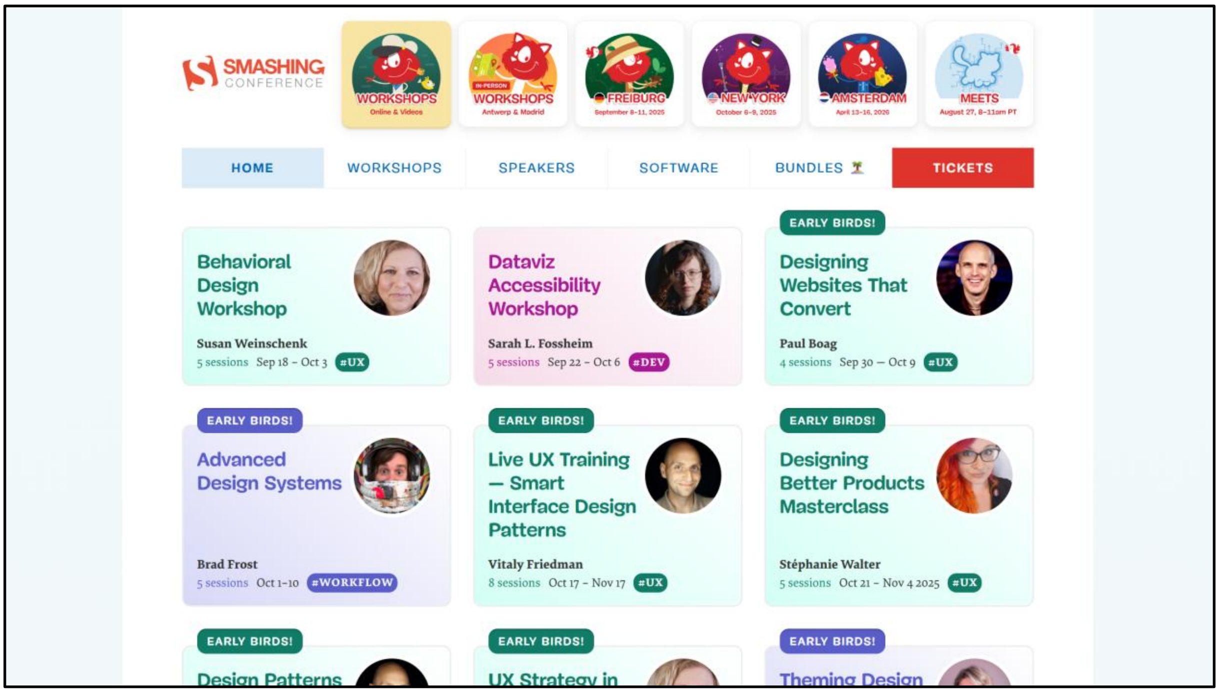
DECORATING LAYOUTS WITH CSS

My name is Patrick, I work on the Edge team, at Microsoft.
In this short talk, I'll speak about the CSS gap decorations feature on the web.



WHITE SPACE IS EVERYWHERE

When we talk about white space in design, we’re talking about any blank or unmarked space between visual elements. It’s also known as negative space, and it plays a critical role in how design is perceived and experienced.
White space exists everywhere: around text, between images, within margins, between lines of text.


SMASHING CONFERENCE
WORKSHOPS Online & Videos
IN-PERSON WORKSHOPS Antwerp & Madrid
FREIBURG September 8–11, 2025
NEW YORK October 6–9, 2025
AMSTERDAM April 13–16, 2026
MEETS August 27, 8–11am PT
HOME
WORKSHOPS
SPEAKERS
SOFTWARE
BUNDLES
TICKETS

Behavioral Design Workshop
Susan Weinschenk
5 sessions Sep 18 – Oct 3 #UX

Dataviz Accessibility Workshop
Sarah L. Fossheim
5 sessions Sep 22 – Oct 6 #DEV

EARLY BIRDS!
Designing Websites That Convert
Paul Boag
4 sessions Sep 30 — Oct 9 #UX

EARLY BIRDS!
Advanced Design Systems
Brad Frost
5 sessions Oct 1–10 #WORKFLOW

EARLY BIRDS!
Live UX Training — Smart Interface Design Patterns
Vitaly Friedman
8 sessions Oct 17 – Nov 17 #UX

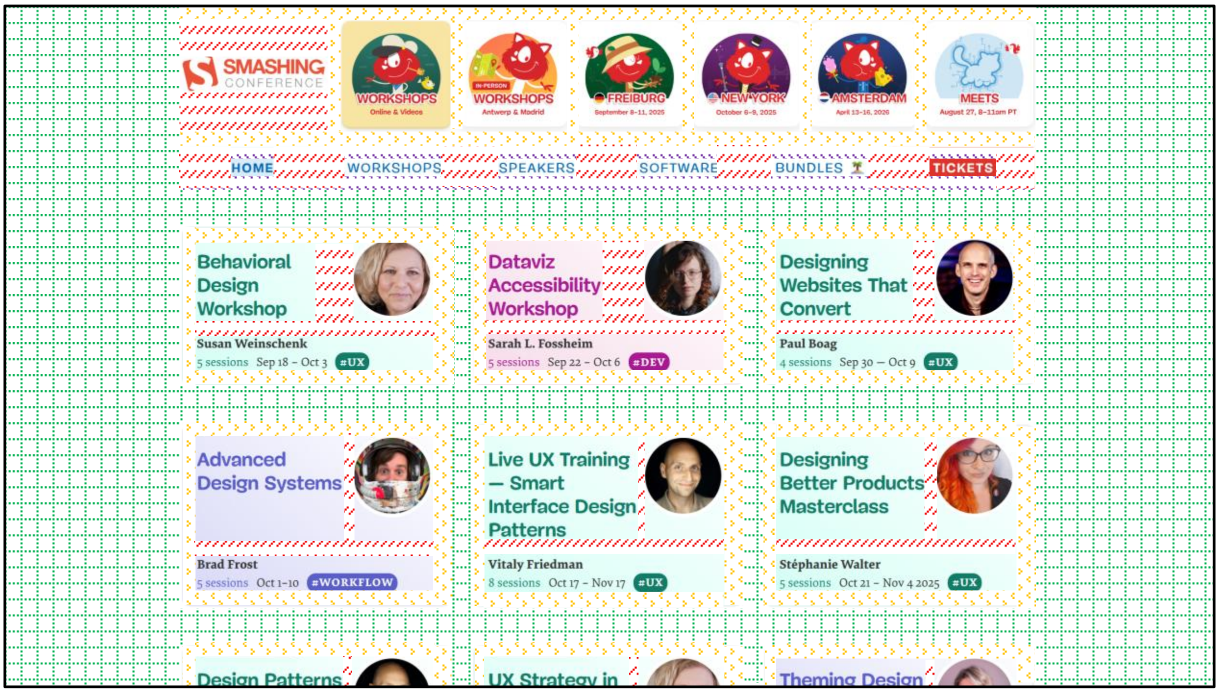
EARLY BIRDS!
Designing Better Products Masterclass
Stéphanie Walter
5 sessions Oct 21 – Nov 4 2025 #UX

EARLY BIRDS!
Design Patterns

EARLY BIRDS!
UX Strategy in

EARLY BIRDS!
Theming Design

Let's highlight the white space on the smashing conf site.


SMASHING CONFERENCE
WORKSHOPS Online & Videos
IN-PERSON WORKSHOPS Antwerp & Madrid
FREIBURG September 8–11, 2025
NEW YORK October 6–9, 2025
AMSTERDAM April 13–16, 2026
MEETS August 27, 8–11am PT
HOME
WORKSHOPS
SPEAKERS
SOFTWARE
BUNDLES
TICKETS

Behavioral Design Workshop
Susan Weinschenk
5 sessions Sep 18 – Oct 3 #UX

Dataviz Accessibility Workshop
Sarah L. Fossheim
5 sessions Sep 22 – Oct 6 #DEV

Designing Websites That Convert
Paul Boag
4 sessions Sep 30 – Oct 9 #UX

Advanced Design Systems
Brad Frost
5 sessions Oct 1–10 #WORKFLOW

Live UX Training — Smart Interface Design Patterns
Vitaly Friedman
8 sessions Oct 17 – Nov 17 #UX

Designing Better Products Masterclass
Stéphanie Walter
5 sessions Oct 21 – Nov 4 2025 #UX

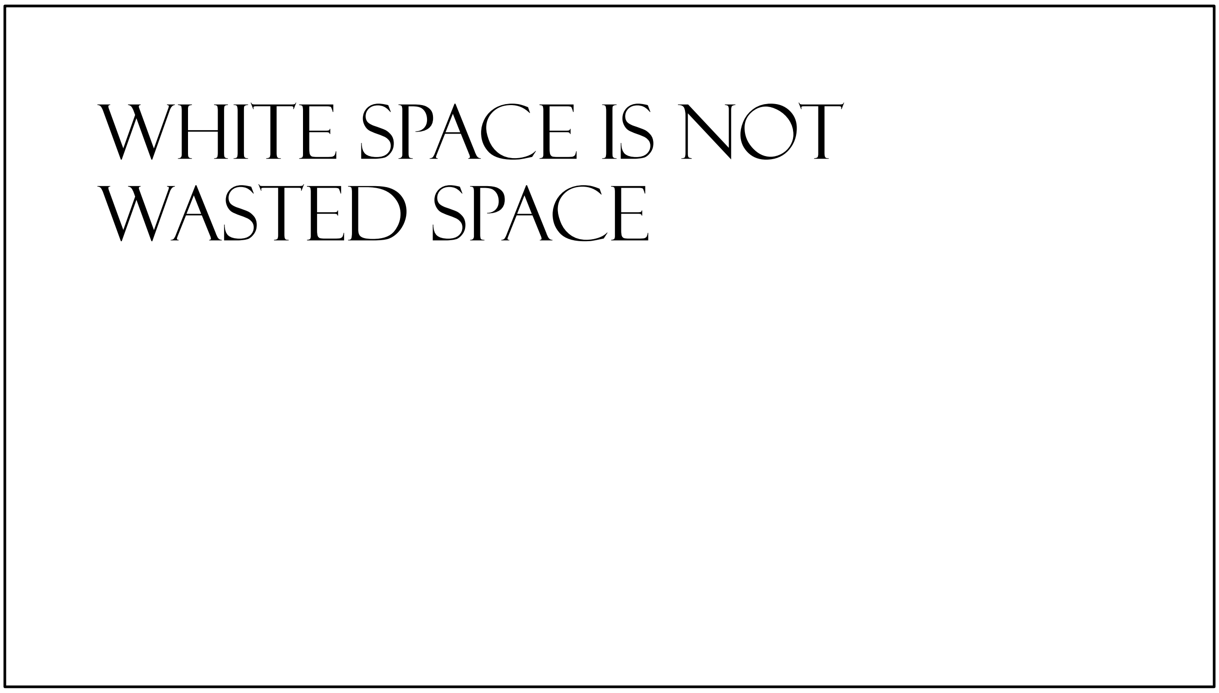
Design Patterns

UX Strategy in

Theming Design

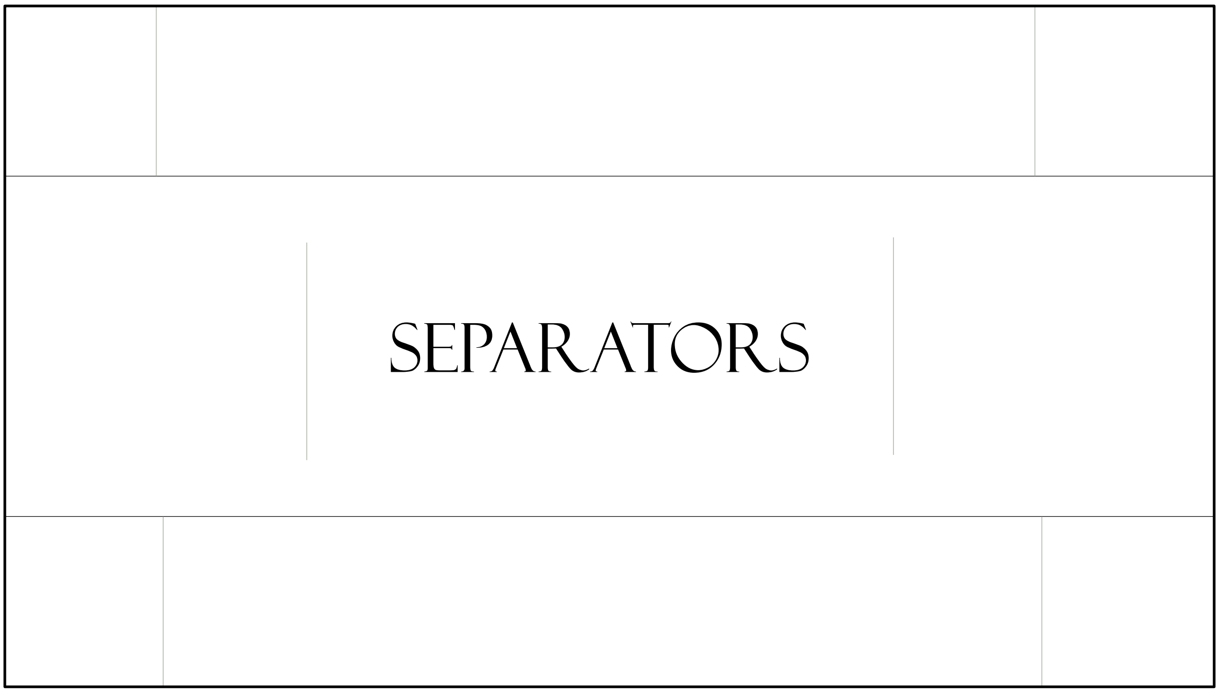
In a lot of different places.
Different types of white spaces.

It's often divided into two types: macro white space, which refers to the larger areas around major elements like blocks of text or images, and micro white space, which includes the small gaps between lines, letters, or buttons.



WHITE SPACE IS NOT WASTED SPACE

White space is far from wasted room.
One of the most powerful tools in a designer’s toolbox.
Creates visual hierarchy, improves readability, guides the viewer’s attention.
Makes content feel open, balanced, and elegant.

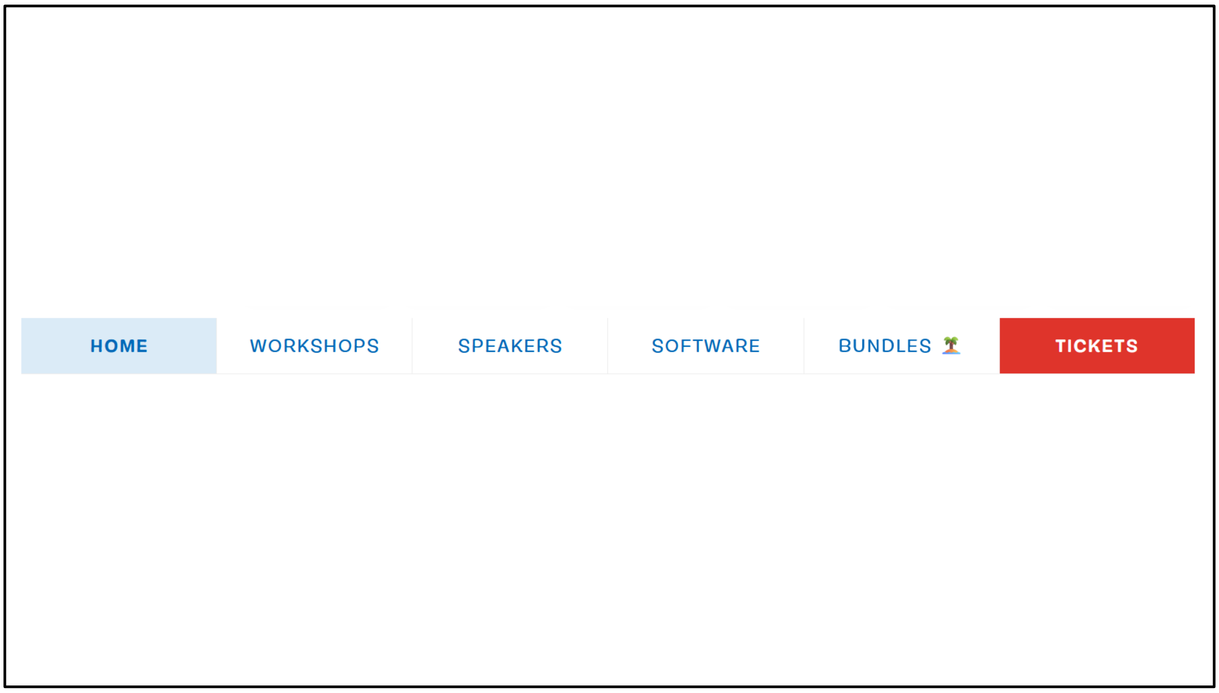


SEPARATORS

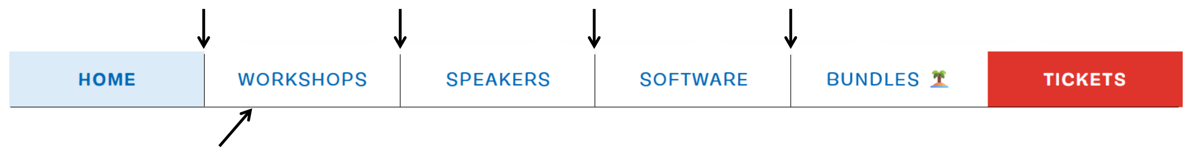
Now, there are also times when white space is not enough on its own
That’s where visual separator lines come in.

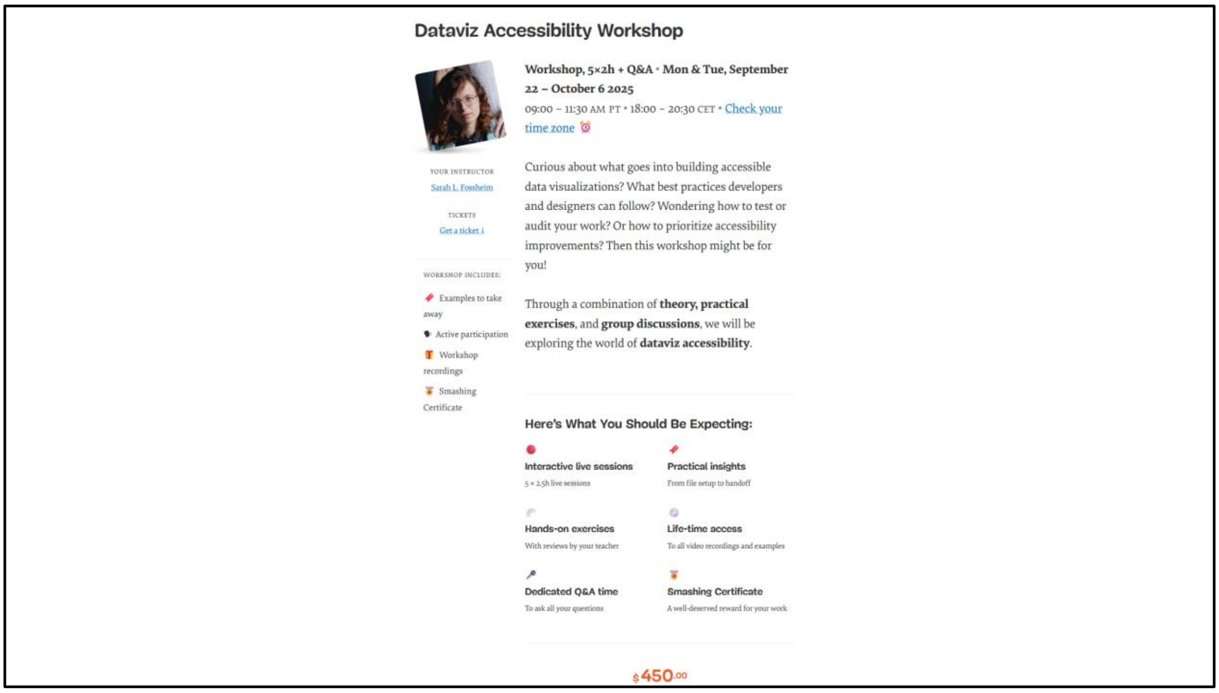
Separator lines (horizontal rules, dividing borders, …) add clear visual break between sections
when white space alone might feel too subtle or ambiguous.

Define structure, improve scannability, guide the eye
Especially in dense interfaces like dashboards, menus, or long-form content.



Let’s take a look at smashing conf again. The top navigation uses separators.
Very subtle, but they’re here and play an important role to disambiguate between the various nav items.
Let’s highlight them.


HOME
WORKSHOPS
SPEAKERS
SOFTWARE
BUNDLES 🏝
TICKETS



Dataviz Accessibility Workshop

Workshop, 5×2h + Q&A · Mon & Tue, September 22 – October 6 2025
09:00 – 11:30 AM PT • 18:00 – 20:30 CET • Check your time zone ⏰

YOUR INSTRUCTOR
Sarah L. Fossheim

TICKETS
Get a ticket ↓

WORKSHOP INCLUDES:
- Examples to take away
- Active participation
- Workshop recordings
- Smashing Certificate

Curious about what goes into building accessible data visualizations? What best practices developers and designers can follow? Wondering how to test or audit your work? Or how to prioritize accessibility improvements? Then this workshop might be for you!

Through a combination of **theory, practical exercises**, and **group discussions**, we will be exploring the world of **dataviz accessibility**.

Here's What You Should Be Expecting:

Interactive live sessions
5 × 2.5h live sessions

Practical insights
From file setup to handoff

Hands-on exercises
With reviews by your teacher

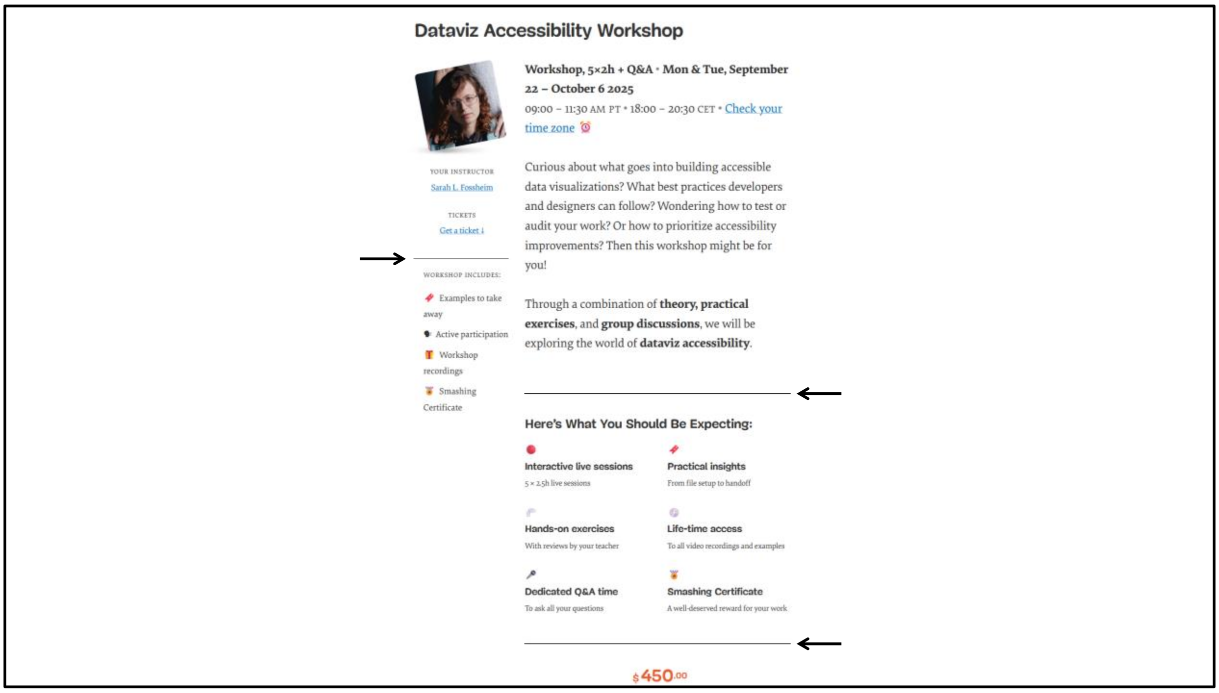
Life-time access
To all video recordings and examples

Dedicated Q&A time
To ask all your questions

Smashing Certificate
A well-deserved reward for your work

$450.00

On a workshop description page too. Also subtle, but between the various sections of the page.
Highlight them.



Dataviz Accessibility Workshop

YOUR INSTRUCTOR
Sarah L. Fossheim

TICKETS
Get a ticket ↓

WORKSHOP INCLUDES:

- Examples to take away
- Active participation
- Workshop recordings
- Smashing Certificate

Workshop, 5×2h + Q&A • Mon & Tue, September 22 – October 6 2025
09:00 – 11:30 AM PT • 18:00 – 20:30 CET • Check your time zone

Curious about what goes into building accessible data visualizations? What best practices developers and designers can follow? Wondering how to test or audit your work? Or how to prioritize accessibility improvements? Then this workshop might be for you!

Through a combination of **theory, practical exercises**, and **group discussions**, we will be exploring the world of **dataviz accessibility**.

Here's What You Should Be Expecting:

Interactive live sessions
5 × 2.5h live sessions

Practical insights
From file setup to handoff

Hands-on exercises
With reviews by your teacher

Life-time access
To all video recordings and examples

Dedicated Q&A time
To ask all your questions

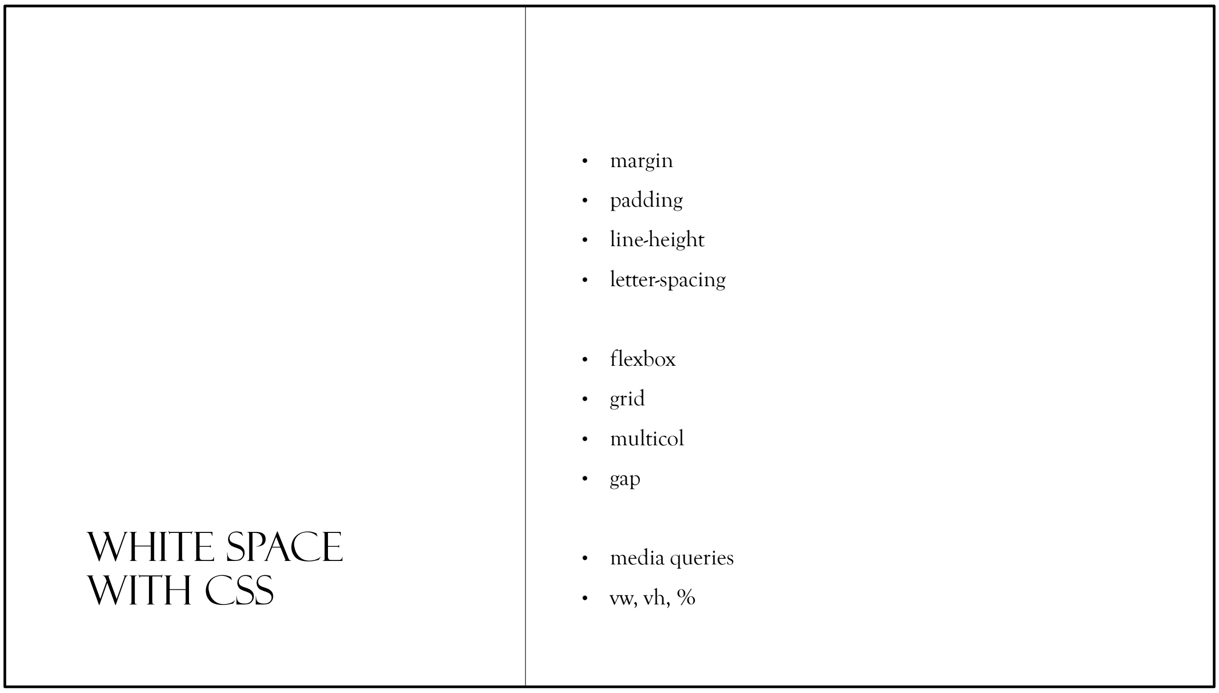
Smashing Certificate
A well-deserved reward for your work

$450.00

On the Web?

What makes the web truly unique is its adaptability.
The web adapts to everyone and every device.
That's its beauty.

At least if you use the right tools for the job.
And for white space and separators, CSS is the right tool.
You could use HTML to add more elements, or JS to place them.
But you’d have to do a lot of work, which could more simply be done in CSS, in a way that’s natively adaptable.



WHITE SPACE WITH CSS

- margin
- padding
- line-height
- letter-spacing

- flexbox
- grid
- multicol
- gap

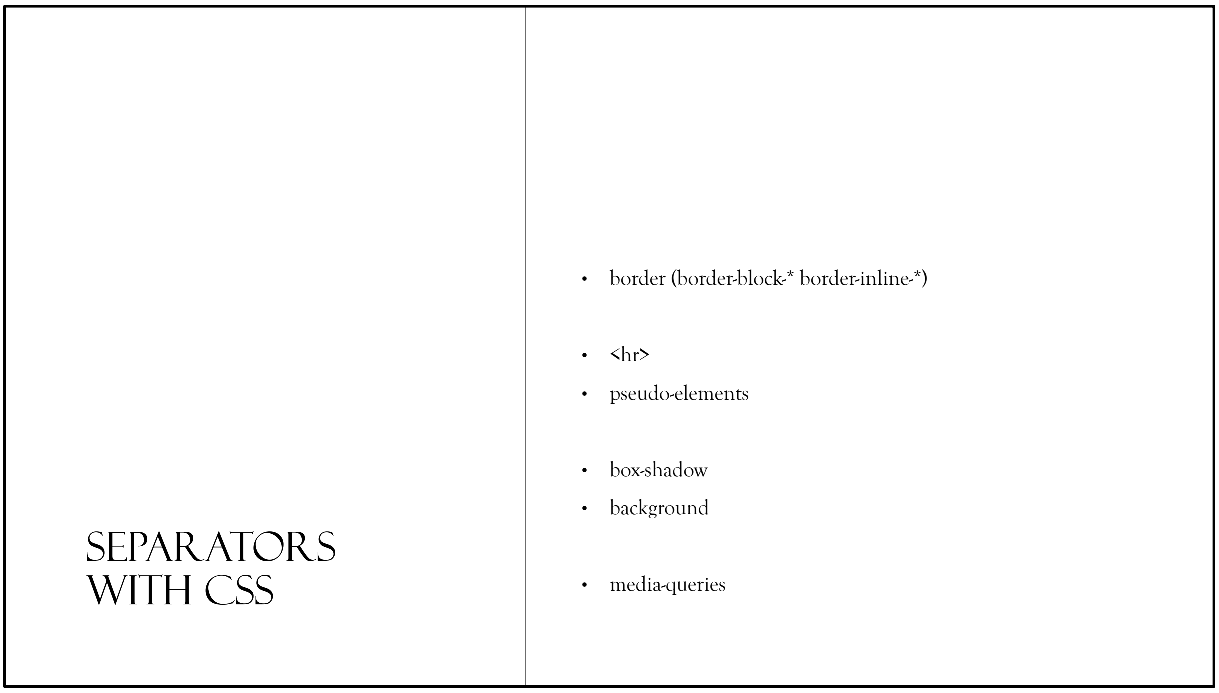
- media queries
- vw, vh, %

CSS gives you incredible control over white space on the web.
And in a way that's responsive to different screen sizes and devices.

properties like margin, padding, gap, and line-height, you can shape the space around and within elements with precision.

And with flexbox and grid, CSS also introduces responsive spacing, without the need for extra HTML.
Using gap to make it easy to manage consistent space between items.

You can use media queries to adjust white space based on screen size, or even use units like vw, vh, and percentages to scale space proportionally.



SEPARATORS WITH CSS

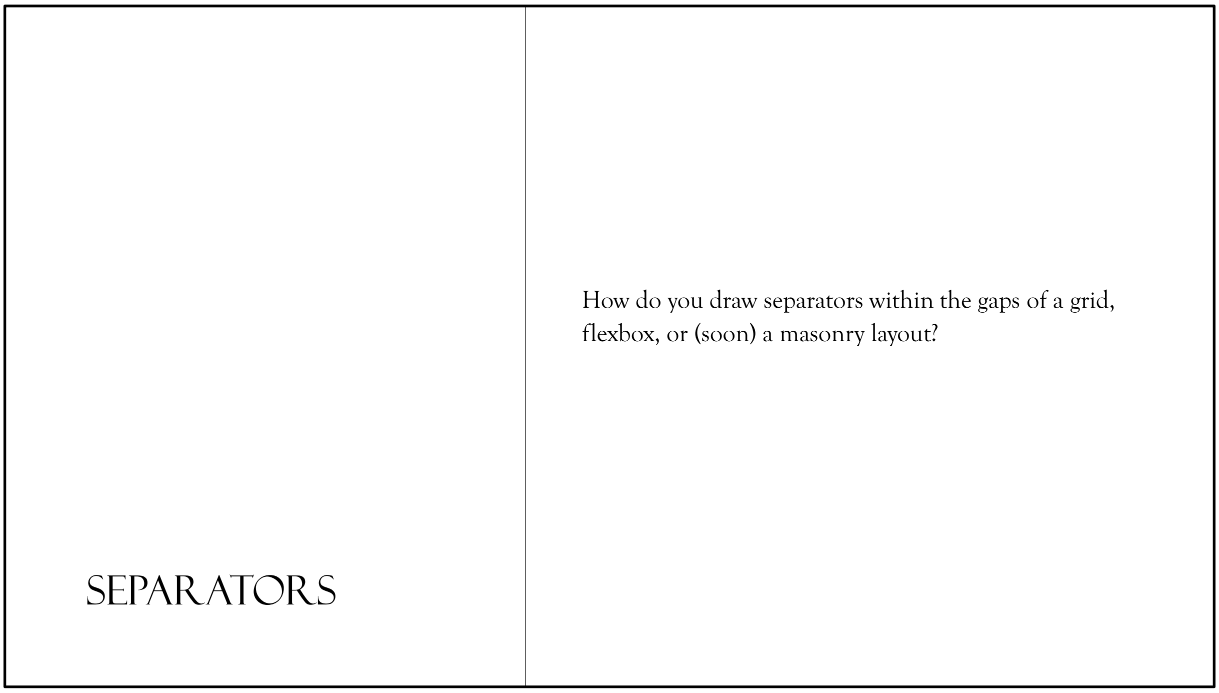
- border (border-block-* border-inline-*)
- <hr>
- pseudo-elements
- box-shadow
- background
- media-queries

CSS also gives you precise control over separator lines, letting you create clean, effective dividers that work beautifully alongside white space.

The most common tool is the border property of course. Even better when you use its logical long-hands, for localization.

You can also use the <hr> HTML tag, which CSS can style with properties like height, color, opacity, or even gradients to match your brand's look.
Same with pseudo-elements, when you don't need an element to pollute your nice semantic markup.

For more flexibility, box-shadow and background can be used creatively to create custom separator effects without relying on actual lines.
Importantly, these separators can be made responsive. With media queries, you can adjust their thickness, visibility, or remove them entirely on smaller screens where space is limited.

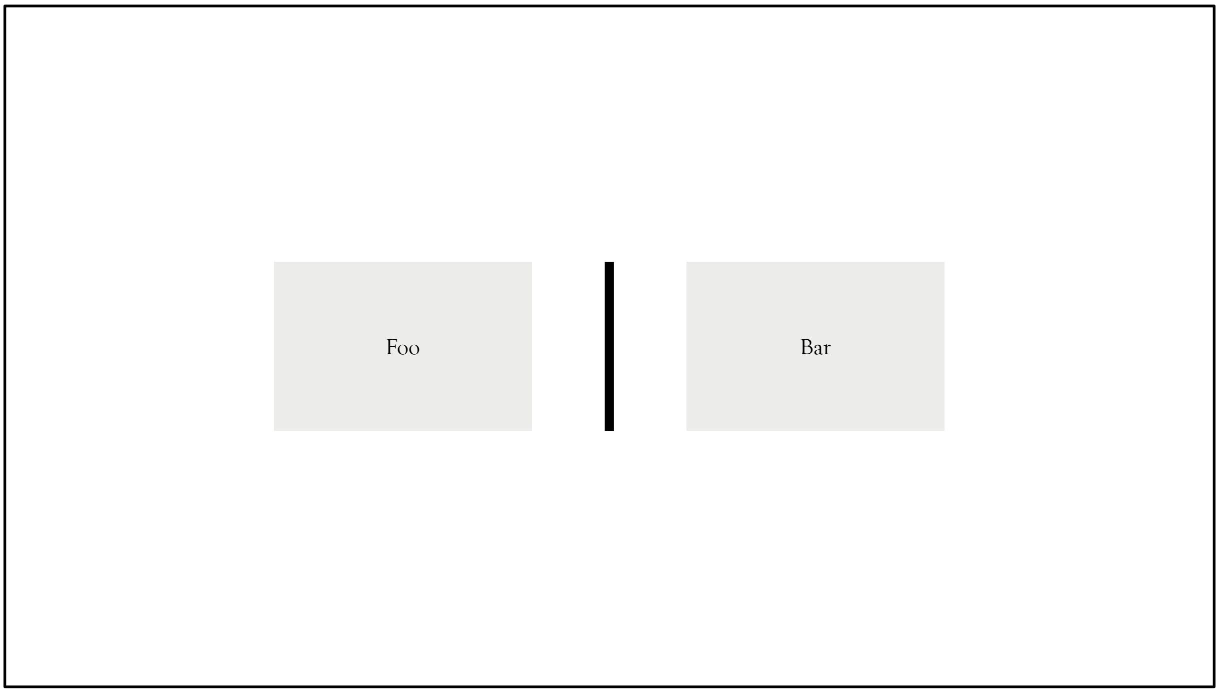


SEPARATORS

How do you draw separators within the gaps of a grid, flexbox, or (soon) a masonry layout?

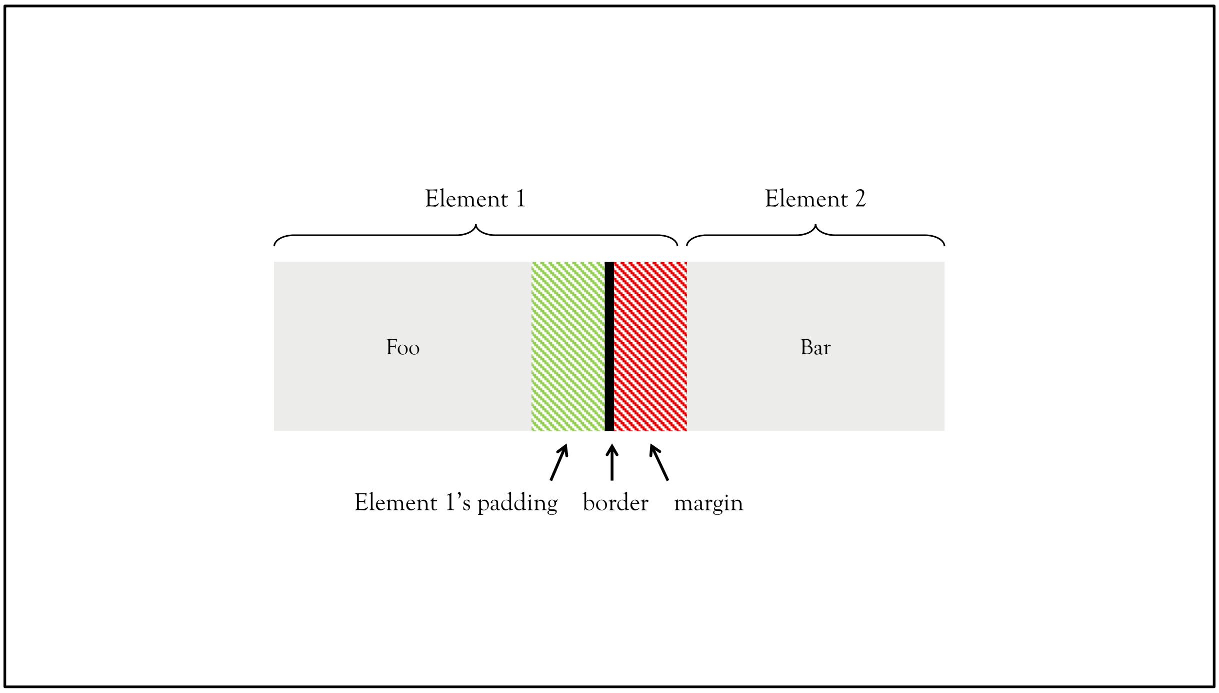
Flexbox, grid, multicol, and soon masonry have become the de-facto tools for layout on the web.
They adapt to any viewport size, are very flexible, and hugely powerful.
With minimal code, you can achieve any layout
and let the browser handle the rest.
They provide a lot of design freedom

But drawing separators in these layout modes can be hard.
How do you add a separator in between 2 cells of a grid layout that has gaps?

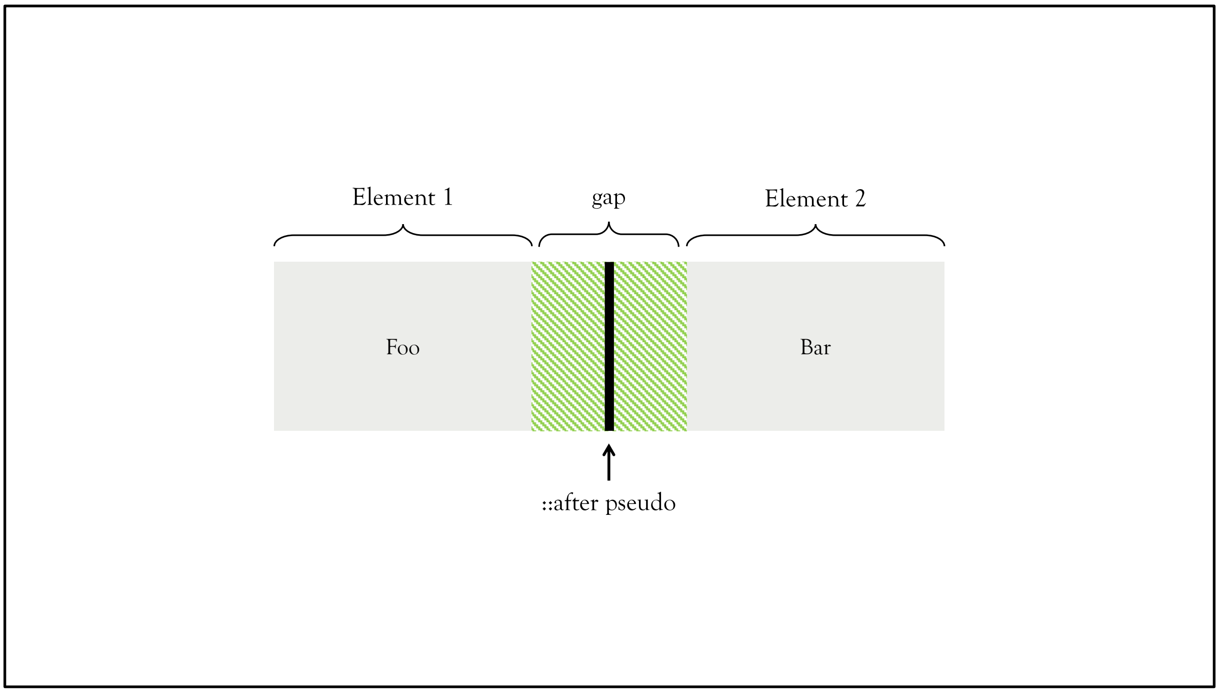


Foo

Bar



Add a border? Need to space it out with padding and margin.
Can't use gap anymore.
How do you not add it to the right of element 2? If it’s responsive, you don’t know how many items there will be, and how wide the container will be.



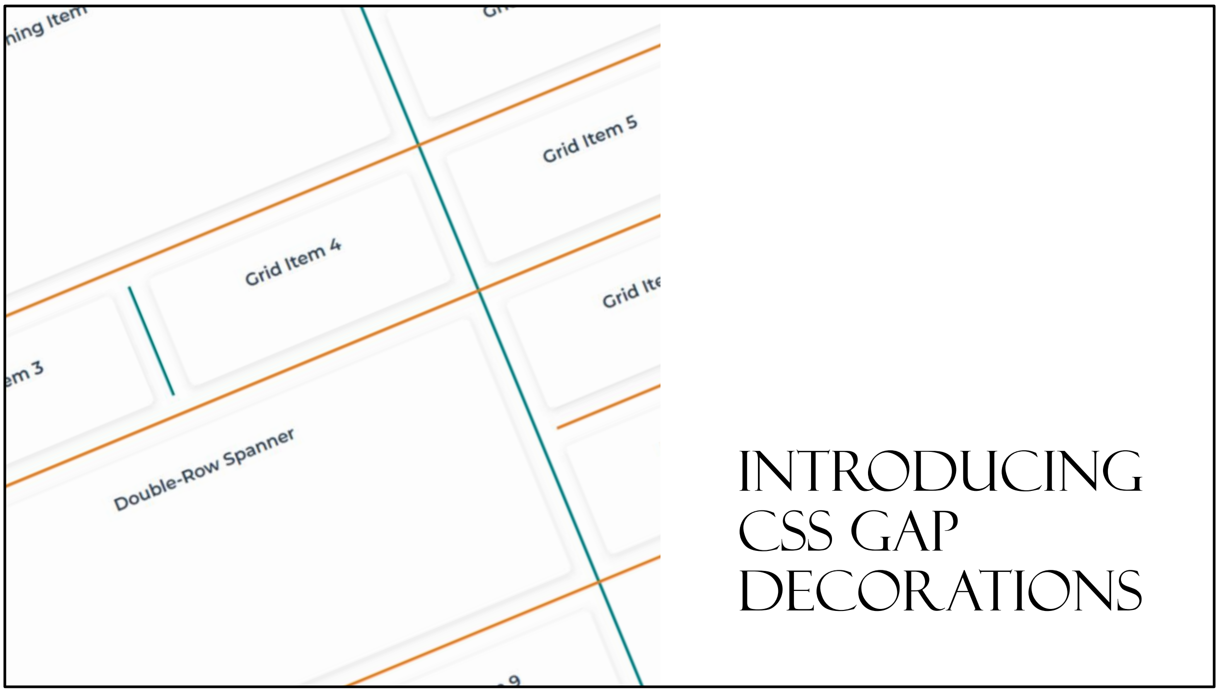
Add a pseudo-element? A bit better.
But same problem as before with not knowing where to add them and when not to.

You could also add extra elements to your market: but not semantic, hard to make responsive.

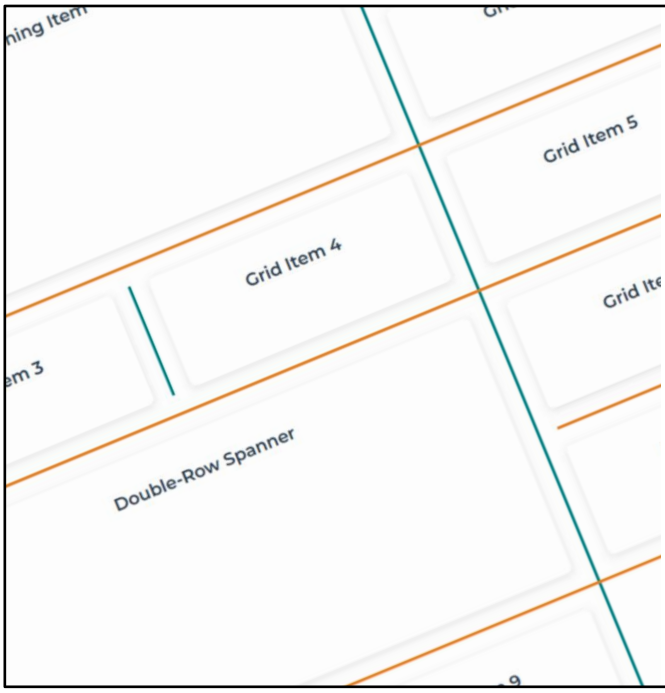
There are other workarounds. But they all come with limitations and challenges:

Unintuitive: They introduce structural dependencies for visual styling, which goes against the principles of semantic HTML.
Accessibility-unfriendly: They often require extra DOM elements, which could interfere with assistive technologies, like screen readers.
Difficult to maintain: They require complicated layout logic and make consistent styling across components more difficult.
Bad for performance: These workarounds may add unnecessary elements to the DOM which can lead to performance issues.

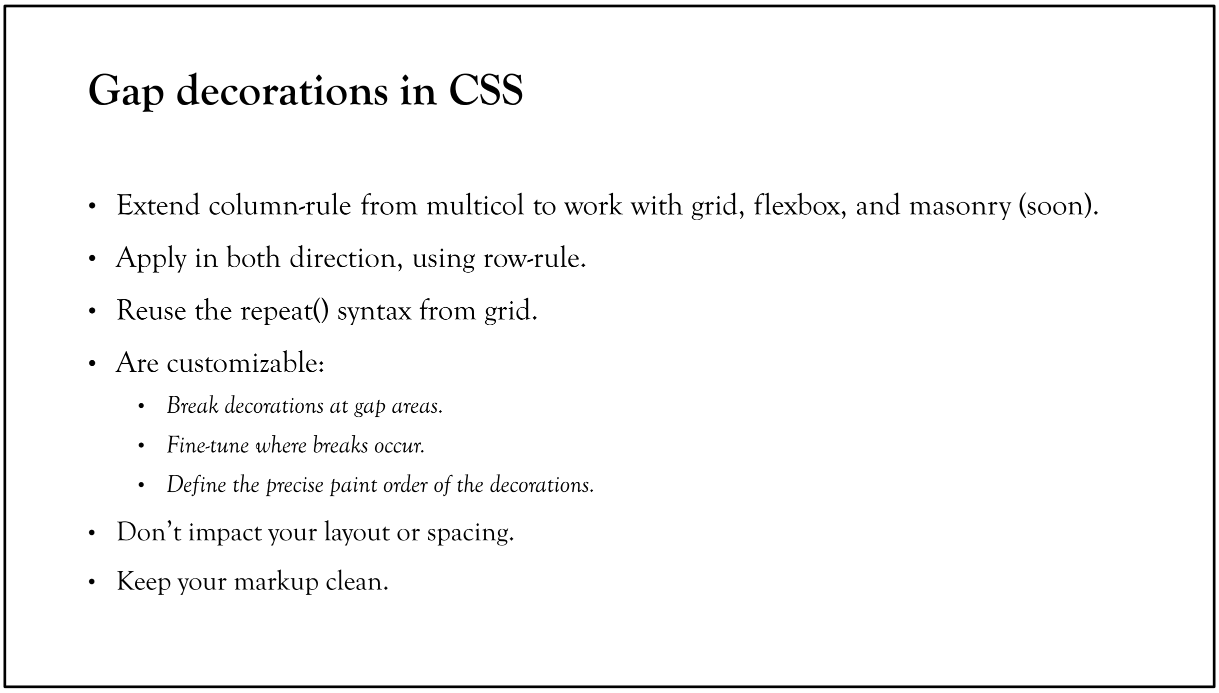
We need a simple way to do these subtle separators. As simple as gap is.



INTRODUCING CSS GAP DECORATIONS



Introducing css gap decorations.
A new proposal we, on the Microsoft Edge team, are working on
And implementing in Chromium-based browsers: Edge and Chrome, and others.



Gap decorations in CSS

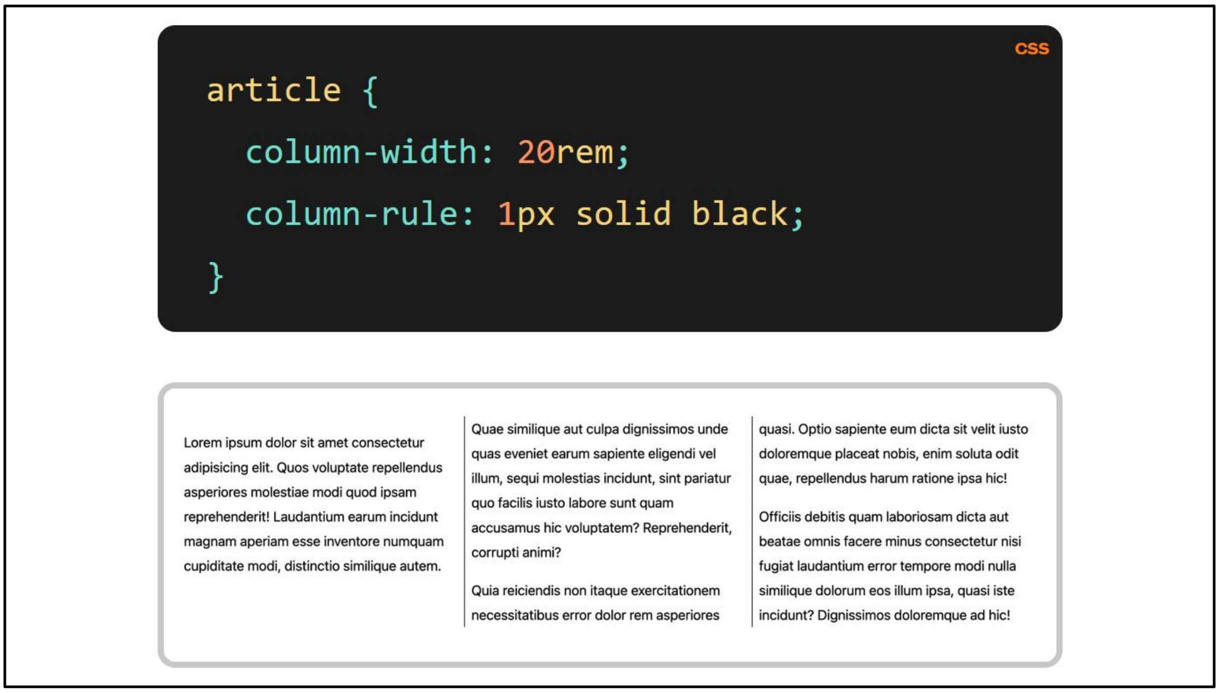
- Extend column-rule from multicol to work with grid, flexbox, and masonry (soon).
- Apply in both direction, using row-rule.
- Reuse the repeat() syntax from grid.
- Are customizable:
  - *Break decorations at gap areas.*
  - *Fine-tune where breaks occur.*
  - *Define the precise paint order of the decorations.*
- Don't impact your layout or spacing.
- Keep your markup clean.

Gap decorations:

extend the concept of column-rule that's in multicol, to other layouts.
also work in both directions, with row-rule.
reuse the same repeat() syntax which grid has, so you can define your decorations in a simple and responsive way.
are super flexible.

This provides you with a solution to draw separators in a way that:

Doesn't impact your layout: purely visual. No impact on layout or spacing, so you can adopt them without fear of breaking existing designs.
Keeps your markup clean: no need for extra elements or pseudo-elements.
Is highly customizable



CSS

```css
article {
  column-width: 20rem;
  column-rule: 1px solid black;
}
```

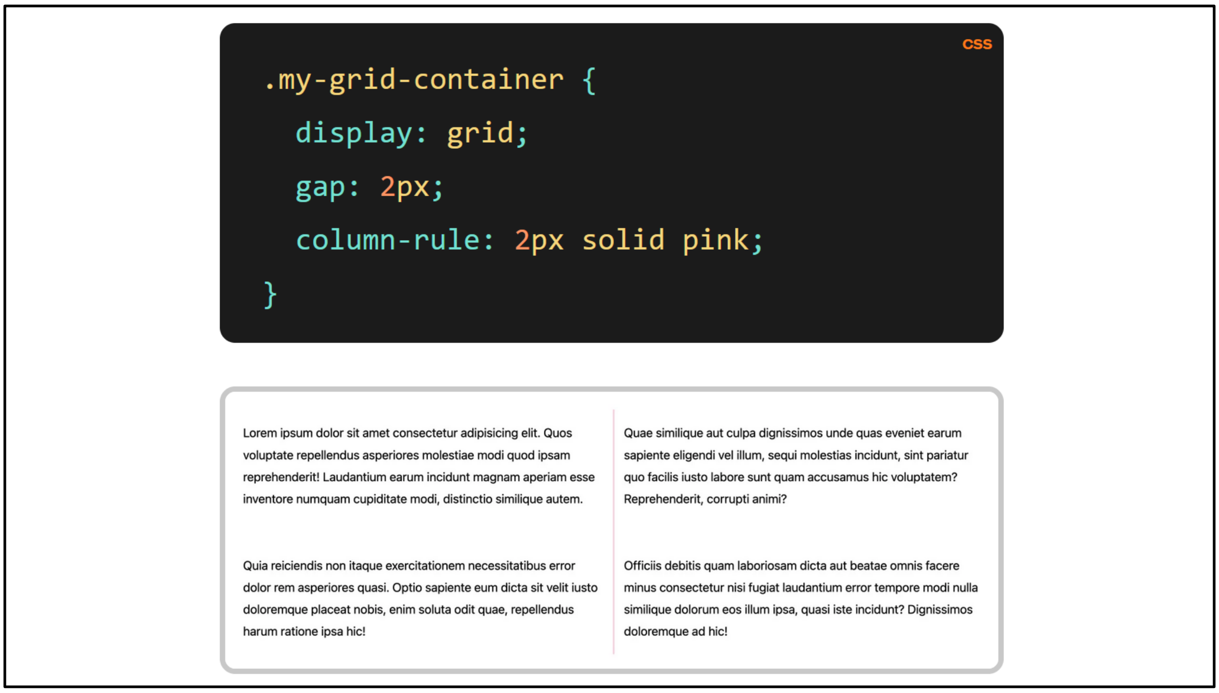
Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptate repellendus asperiores molestiae modi quod ipsam reprehenderit! Laudantium earum incidunt magnam aperiam esse inventore numquam cupiditate modi, distinctio similique autem.

Quae similique aut culpa dignissimos unde quas eveniet earum sapiente eligendi vel illum, sequi molestias incidunt, sint pariatur quo facilis iusto labore sunt quam accusamus hic voluptatem? Reprehenderit, corrupti animi?

Quia reiciendis non itaque exercitationem necessitatibus error dolor rem asperiores quasi. Optio sapiente eum dicta sit velit iusto doloremque placeat nobis, enim soluta odit quae, repellendus harum ratione ipsa hic!

Officiis debitis quam laboriosam dicta aut beatae omnis facere minus consectetur nisi fugiat laudantium error tempore modi nulla similique dolorum eos illum ipsa, quasi iste incidunt? Dignissimos doloremque ad hic!

```
article {
  column-width: 20rem;
  column-rule: 1px solid black;
}
```



css

```
.my-grid-container {
  display: grid;
  gap: 2px;
  column-rule: 2px solid pink;
}
```

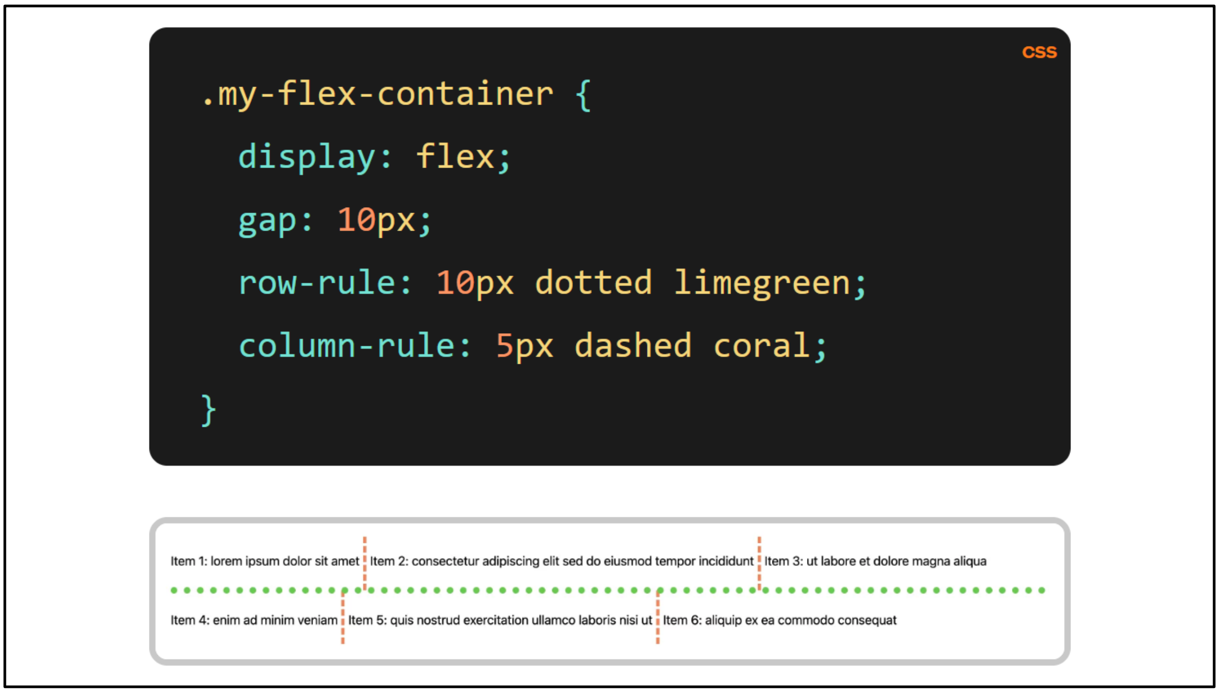
Lorem ipsum dolor sit amet consectetur adipisicing elit. Quos voluptate repellendus asperiores molestiae modi quod ipsam reprehenderit! Laudantium earum incidunt magnam aperiam esse inventore numquam cupiditate modi, distinctio similique autem.

Quae similique aut culpa dignissimos unde quas eveniet earum sapiente eligendi vel illum, sequi molestias incidunt, sint pariatur quo facilis iusto labore sunt quam accusamus hic voluptatem? Reprehenderit, corrupti animi?

Quia reiciendis non itaque exercitationem necessitatibus error dolor rem asperiores quasi. Optio sapiente eum dicta sit velit iusto doloremque placeat nobis, enim soluta odit quae, repellendus harum ratione ipsa hic!

Officiis debitis quam laboriosam dicta aut beatae omnis facere minus consectetur nisi fugiat laudantium error tempore modi nulla similique dolorum eos illum ipsa, quasi iste incidunt? Dignissimos doloremque ad hic!

```
.my-grid-container {
  display: grid;
  gap: 2px;
  column-rule: 2px solid pink;
}
```

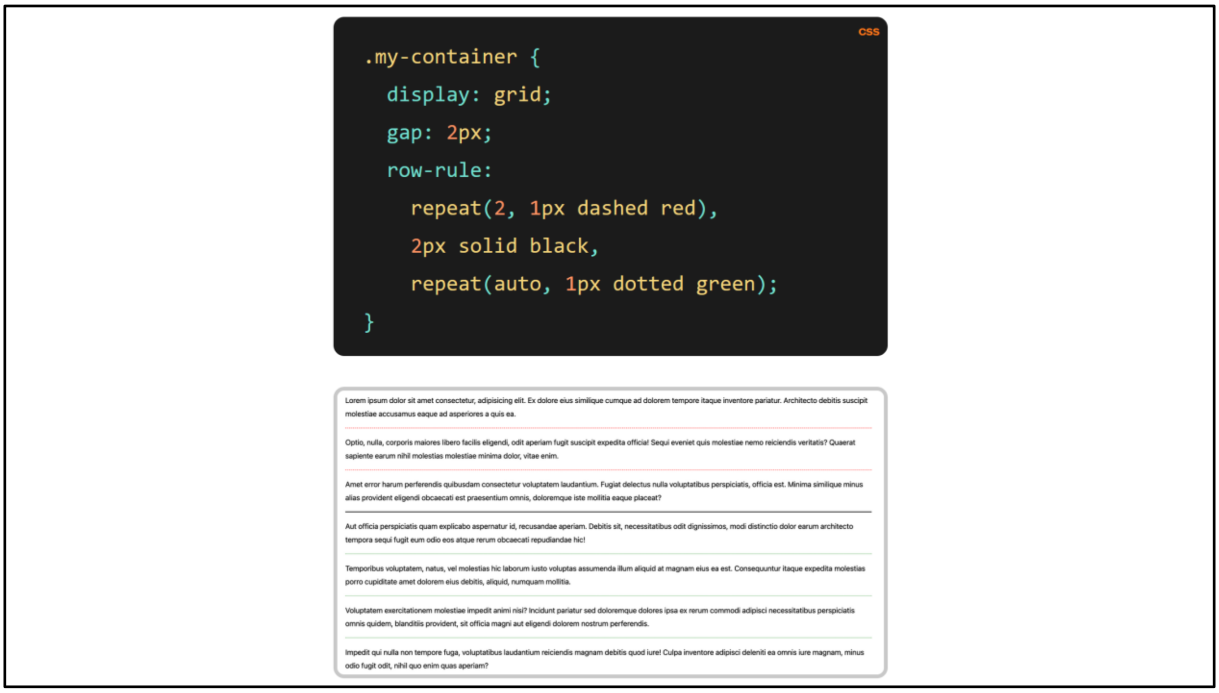


```css
.my-flex-container {
  display: flex;
  gap: 10px;
  row-rule: 10px dotted limegreen;
  column-rule: 5px dashed coral;
}
```

Item 1: lorem ipsum dolor sit amet | Item 2: consectetur adipiscing elit sed do eiusmod tempor incididunt | Item 3: ut labore et dolore magna aliqua

Item 4: enim ad minim veniam | Item 5: quis nostrud exercitation ullamco laboris nisi ut | Item 6: aliquip ex ea commodo consequat

```css
.my-flex-container {
  display: flex;
  gap: 10px;
  row-rule: 10px dotted limegreen;
  column-rule: 5px dashed coral;
}
```



```css
.my-container {
  display: grid;
  gap: 2px;
  row-rule:
    repeat(2, 1px dashed red),
    2px solid black,
    repeat(auto, 1px dotted green);
}
```

Lorem ipsum dolor sit amet consectetur, adipisicing elit. Ex dolore eius similique cumque ad dolorem tempore itaque inventore pariatur. Architecto debitis suscipit molestiae accusamus eaque ad asperiores a quis ea.

Optio, nulla, corporis maiores libero facilis eligendi, odit aperiam fugit suscipit expedita officia! Sequi eveniet quis molestiae nemo reiciendis veritatis? Quaerat sapiente earum nihil molestias molestiae minima dolor, vitae enim.

Amet error harum perferendis quibusdam consectetur voluptatem laudantium. Fugiat delectus nulla voluptatibus perspiciatis, officia est. Minima similique minus alias provident eligendi obcaecati est praesentium omnis, doloremque iste mollitia eaque placeat?

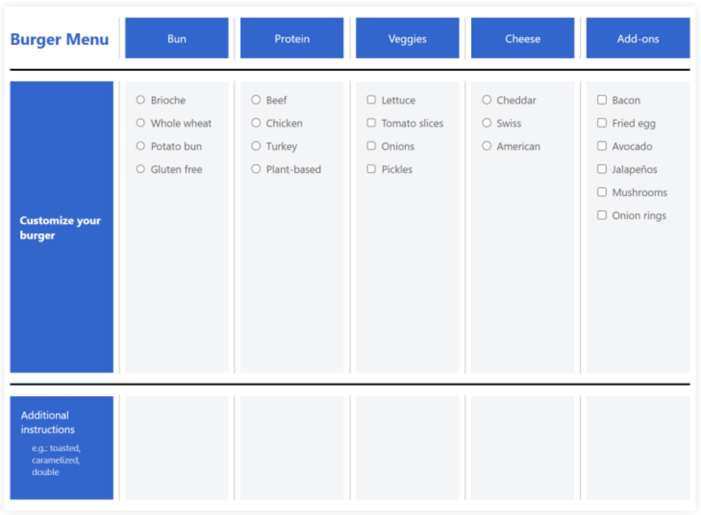
Aut officia perspiciatis quam explicabo aspernatur id, recusandae aperiam. Debitis sit, necessitatibus odit dignissimos, modi distinctio dolor earum architecto tempora sequi fugit eum odio eos atque rerum obcaecati repudiandae hic!

Temporibus voluptatem, natus, vel molestias hic laborum iusto voluptas assumenda illum aliquid at magnam eius ea est. Consequuntur itaque expedita molestias porro cupiditate amet dolorem eius debitis, aliquid, numquam mollitia.

Voluptatem exercitationem molestiae impedit animi nisi? Incidunt pariatur sed doloremque dolores ipsa ex rerum commodi adipisci necessitatibus perspiciatis omnis quidem, blanditiis provident, sit officia magni aut eligendi dolorem nostrum perferendis.

Impedit qui nulla non tempore fuga, voluptatibus laudantium reiciendis magnam debitis quod iure! Culpa inventore adipisci deleniti ea omnis iure magnam, minus odio fugit odit, nihil quo enim quas aperiam?

```css
.my-container {
  display: grid;
  gap: 2px;
  row-rule:
    repeat(2, 1px dashed red),
    2px solid black,
    repeat(auto, 1px dotted green);
}
```

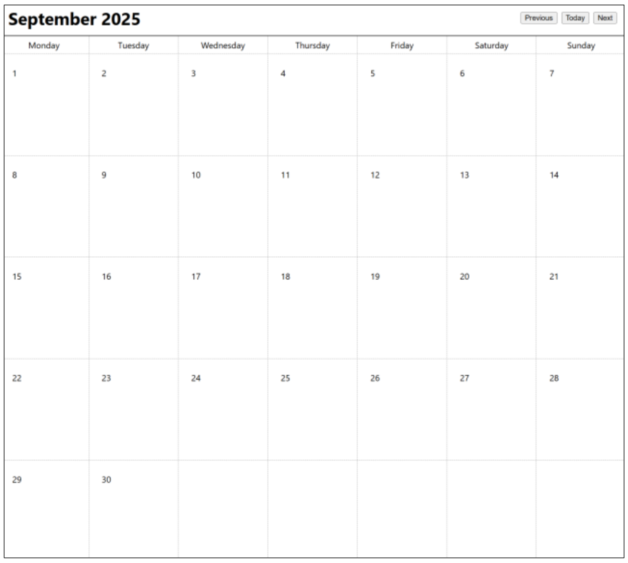


```css
.grid-container {
  display: grid;
  grid-template-columns: repeat(6, 1fr);
  grid-template-rows: 1fr 5fr 2fr;
  gap: 1.5px;

  column-rule: 1.5px solid #ccc;
  row-rule: 3px solid #000;
  column-rule-break: intersection;
  column-rule-outset: -15px;
  row-rule-outset: -8px;
}
```

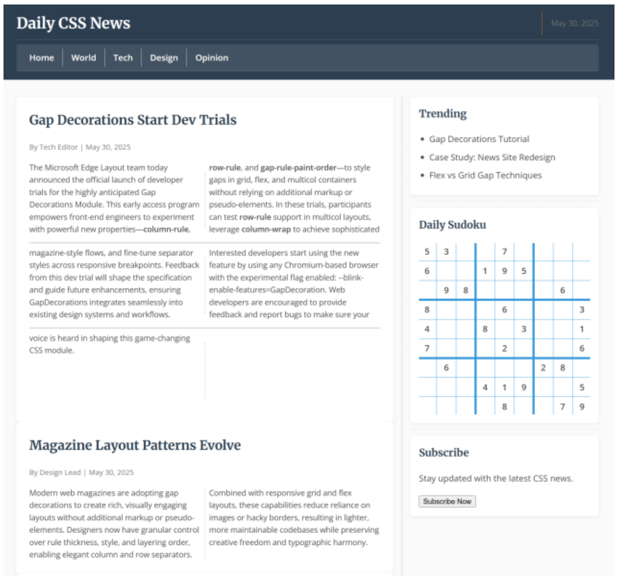
Interesting things:

- Single grid container for everything.
- Column-rules have a break defined, so they only cover the item areas, not the gap, and let the row-rules uncut.



```css
.calendar {
  display: grid;
  grid-template-columns: repeat(7, 1fr);
  grid-template-rows: repeat(6, 1fr);
  gap: 2px;
  row-rule: 1px dashed #999;
  column-rule: 1px dashed #999;
}
```

Subtle separators between these days of a calendar.
Uses grid, super easy with row/column-rule.
Nothing in markup. No hack with background color of calendar.



Daily CSS News

May 30, 2025

Home | World | Tech | Design | Opinion

Gap Decorations Start Dev Trials

By Tech Editor | May 30, 2025

The Microsoft Edge Layout team today announced the official launch of developer trials for the highly anticipated Gap Decorations Module. This early access program empowers front-end engineers to experiment with powerful new properties—**column-rule**, **row-rule**, and **gap-rule-paint-order**—to style gaps in grid, flex, and multicol containers without relying on additional markup or pseudo-elements. In these trials, participants can test **row-rule** support in multicol layouts, leverage **column-wrap** to achieve sophisticated magazine-style flows, and fine-tune separator styles across responsive breakpoints. Feedback from this dev trial will shape the specification and guide future enhancements, ensuring GapDecorations integrates seamlessly into existing design systems and workflows. Interested developers start using the new feature by using any Chromium-based browser with the experimental flag enabled: --blink-enable-features=GapDecoration. Web developers are encouraged to provide feedback and report bugs to make sure your voice is heard in shaping this game-changing CSS module.

Magazine Layout Patterns Evolve

By Design Lead | May 30, 2025

Modern web magazines are adopting gap decorations to create rich, visually engaging layouts without additional markup or pseudo-elements. Designers now have granular control over rule thickness, style, and layering order, enabling elegant column and row separators. Combined with responsive grid and flex layouts, these capabilities reduce reliance on images or hacky borders, resulting in lighter, more maintainable codebases while preserving creative freedom and typographic harmony.

Trending

- Gap Decorations Tutorial
- Case Study: News Site Redesign
- Flex vs Grid Gap Techniques

Daily Sudoku

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Subscribe

Stay updated with the latest CSS news.

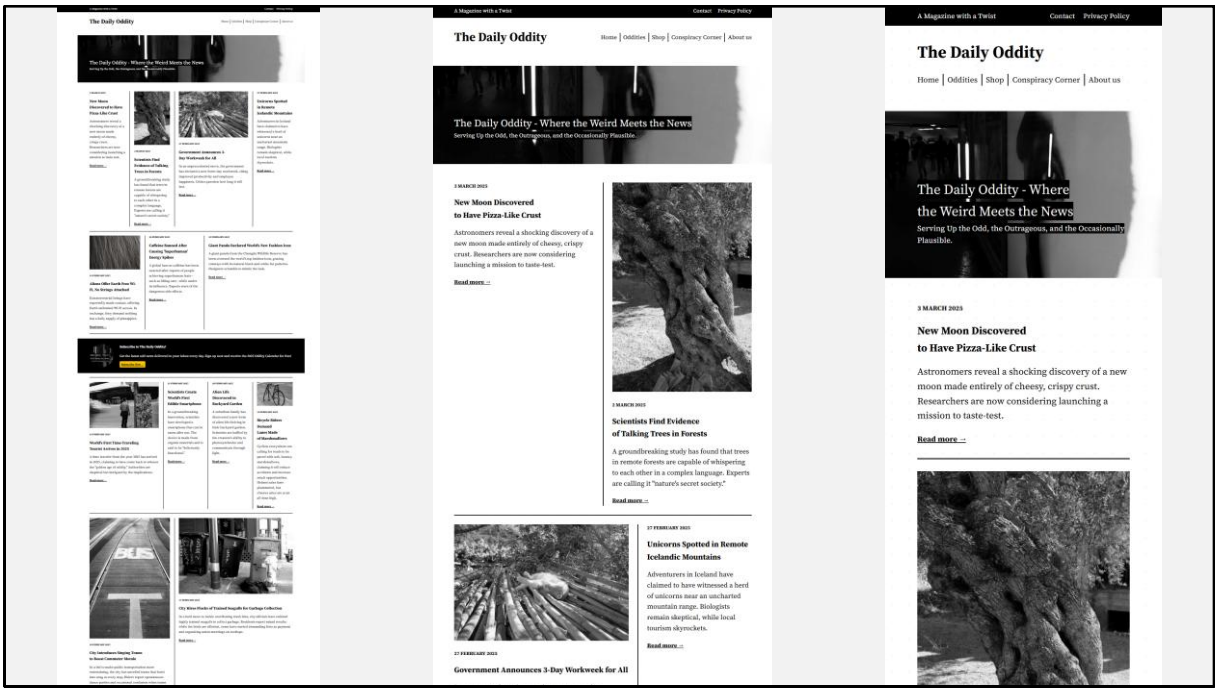
```css
.sudoku {
  display: grid;
  grid-template-columns: repeat(9, 1fr);
  gap: 6px;

  column-rule: 1px solid #3498db;
  row-rule: 1px solid #3498db;

  column-rule-width:
    repeat(2, 1px)
    4px
    repeat(2, 1px)
    4px
    repeat(2, 1px);
  row-rule-width:
    repeat(2, 1px)
    4px
    repeat(2, 1px)
    4px
    repeat(2, 1px);

  gap-rule-paint-order: row-over-column;
}
```

Slightly more complex example of a news site.
See the vertical lines between nav items at the top.
Between columns and rows of content.
Also to separate the sidebar.
Fun fact, the entire sudoku is done with grid and gap decorations.


A Magazine with a Twist

Contact Privacy Policy

The Daily Oddity

Home | Oddities | Shop | Conspiracy Corner | About us

The Daily Oddity - Where the Weird Meets the News

Serving Up the Odd, the Outrageous, and the Occasionally Plausible.

3 MARCH 2025

New Moon Discovered to Have Pizza-Like Crust

Astronomers reveal a shocking discovery of a new moon made entirely of cheesy, crispy crust. Researchers are now considering launching a mission to taste-test.

Read more →

2 MARCH 2025

Scientists Find Evidence of Talking Trees in Forests

A groundbreaking study has found that trees in remote forests are capable of whispering to each other in a complex language. Experts are calling it "nature's secret society."

Read more →

27 FEBRUARY 2025

Unicorns Spotted in Remote Icelandic Mountains

Adventurers in Iceland have claimed to have witnessed a herd of unicorns near an uncharted mountain range. Biologists remain skeptical, while local tourism skyrockets.

Read more →

27 FEBRUARY 2025

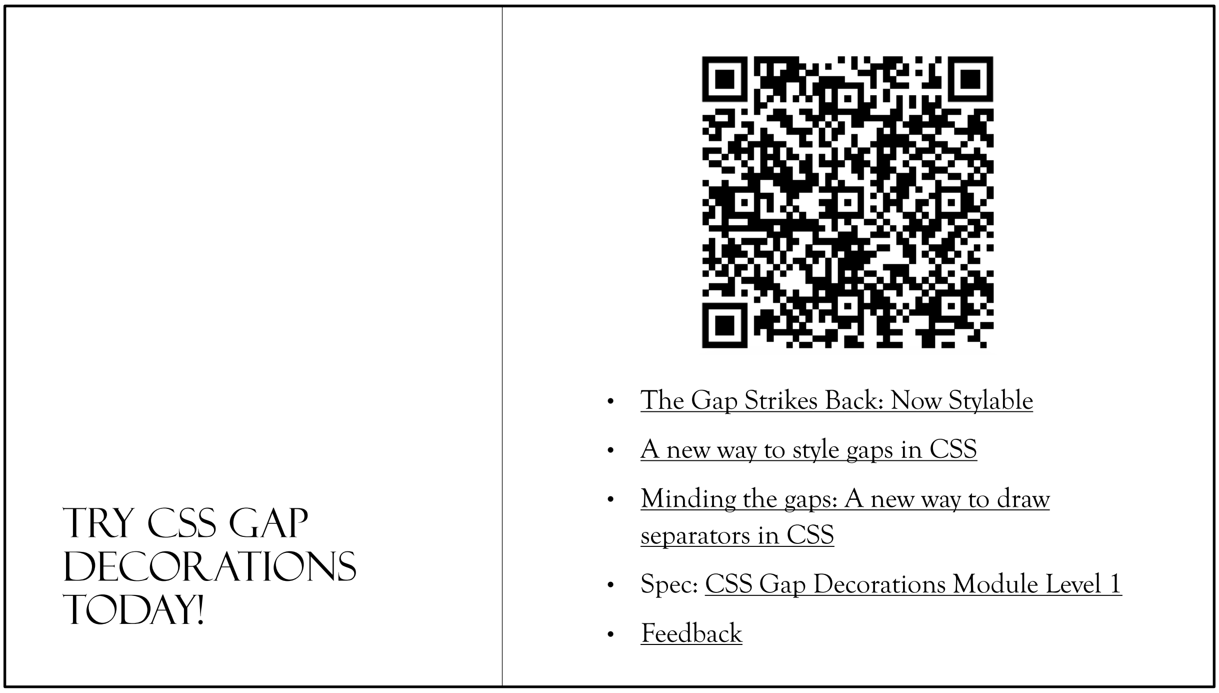
Government Announces 3-Day Workweek for All

Final example, again a news site, where whitespace is key to structuring the layout, but where space is also limited and where therefore clear separators play an important role. They also help reinforce the strong black and white design language here.

The main element, with the articles is a single wrapping flex container.
Articles can be given a variable amount of horizontal space, to emphasize some of them, but also provide an interesting rhythm to the layout.
Some items, like the header, and the ad in the middle can span the entire width of the container.

Gap decorations work seamlessly here. You can customize them to start and end where you want.
Here for instance they leave the full width spanning items alone.

Works beautifully in responsive environments too.
Decorations appear where they're supposed to without you having to do anything.



Try CSS Gap Decorations Today!

- The Gap Strikes Back: Now Stylable
- A new way to style gaps in CSS
- Minding the gaps: A new way to draw separators in CSS
- Spec: CSS Gap Decorations Module Level 1
- Feedback

Try it out today.
Here are all the resources you need to play with the previous demos, and more.
Read articles about the feature.
Read the explainer, the spec.
And more importantly, learn to enable it in Chrome or Edge, play with it.
If you do, please let us know what you think.