

Declarative Design

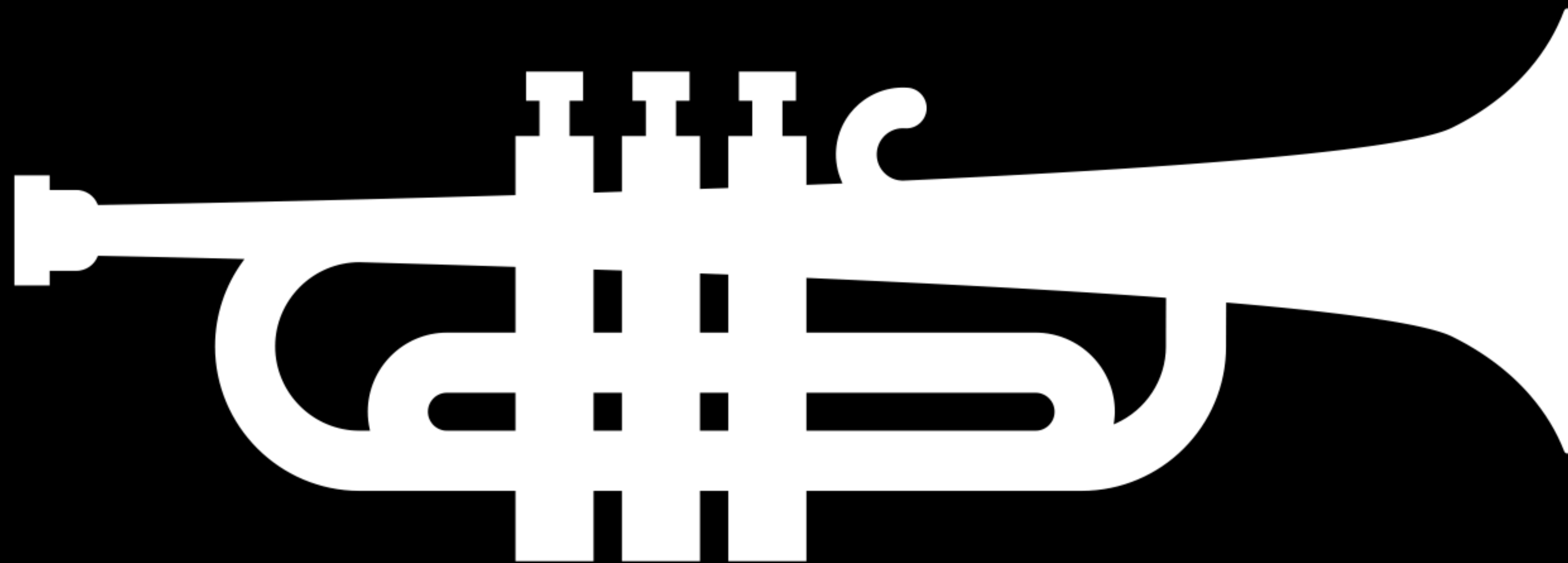












programming

imperative

programming

- 1. create an array of items**
- 2. loop through each item**
- 3. test if a condition is true**
- 4. return the result**

declarative

programming

Structured

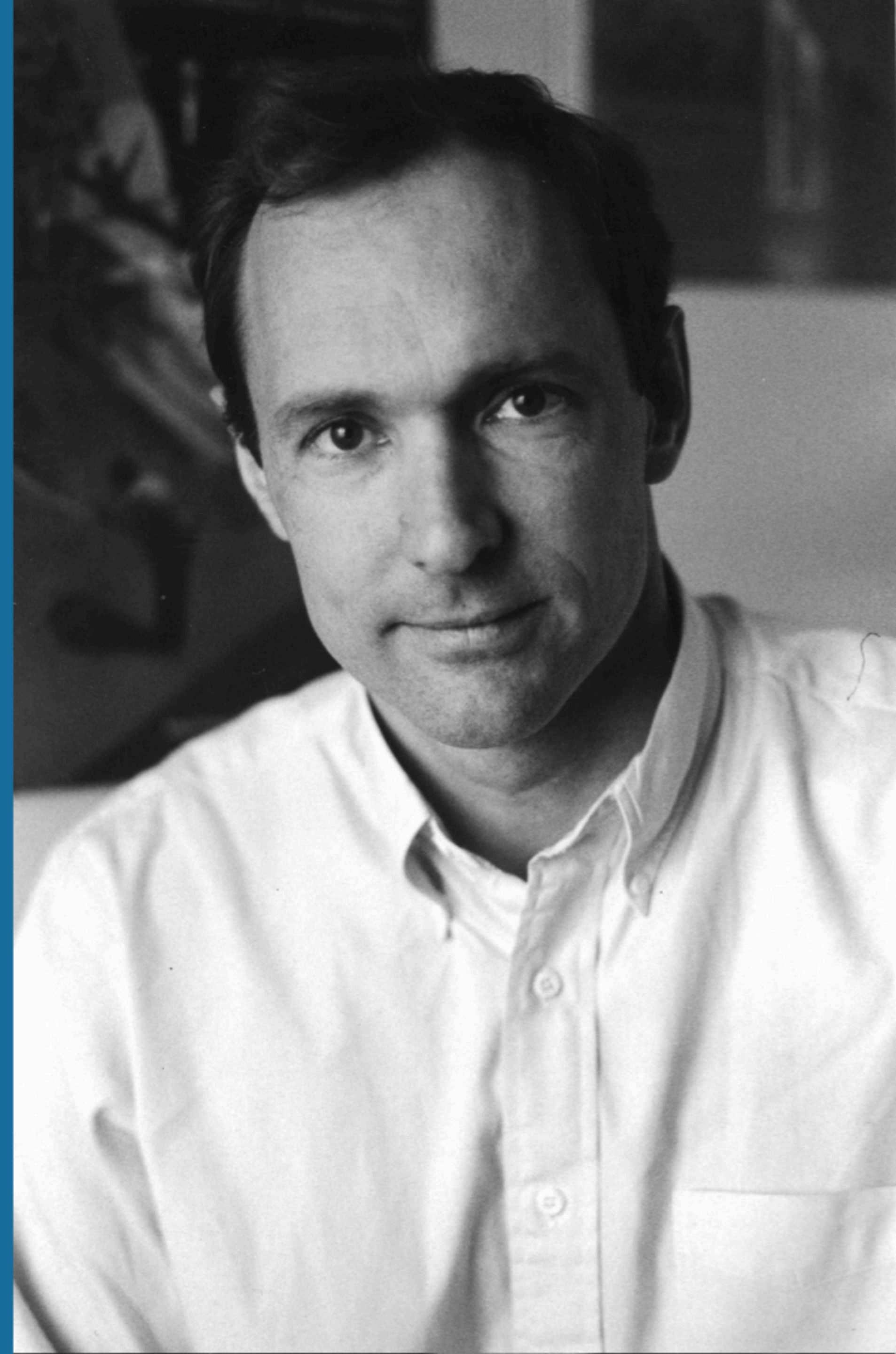
Query

Language

**SELECT items FROM table
WHERE condition IS TRUE**

**World
Wide
Web**

World Wide Web



CERN DD/OC

Tim Berners-Lee, CERN/DD

Information Management: A Proposal

March 1989

Information Management: A Proposal

Abstract

This proposal concerns the management of general information about accelerators and experiments at CERN. It discusses the problems of loss of information about complex evolving systems and derives a solution based on a distributed hypertext system.

Keywords: Hypertext, Computer conferencing, Document retrieval, Information management, Project

The principle of least power

*“Choose the least powerful language
suitable for a given purpose.”*

The principle of least power

HTML

CSS

```
h1.font.size = 24pt 100%  
h2.font.size = 20pt 40%
```

influence

user stylesheets

@prefers-...

JavaScript

JavaScript

CSS

HTML

XML

JavaScript

CSS

HTML

imperative

programming

declarative

imperative

mindset

declarative

button


```
<div
  tabIndex={0}
  onClick={onClickButtonHandler}
  onKeyDown={onKeyDownHandler}
  onKeyUp={onKeyUpHandler}
  role="button"
>
  example
</div>
```

```
function isDisabled(eventTarget) {
  const ariaDisabled = eventTarget.getAttribute("aria-disabled");
  const isDisabled = ariaDisabled === "" || ariaDisabled === "true";

  return isDisabled;
}
```

```
function onKeyDownHandler(event) {
  if (event.key === " ") {
    // We have to disable default space behavior which is scrolling page down.
    event.preventDefault();
  }
  if (event.key === "Enter" && !isDisabled(event.target)) {
    onClickButtonHandler();
  }
}
```

```
function onKeyUpHandler(event) {
  if (event.key === " " && !isDisabled(event.target)) {
    onClickButtonHandler();
  }
}
```

```
<div
  tabIndex={0}
  onClick={onClickButtonHandler}
  onKeyDown={onKeyDownHandler}
  onKeyUp={onKeyUpHandler}
  role="button"
>
  example
</div>
```

```
function isDisabled(eventTarget) {
  const ariaDisabled = eventTarget.getAttribute("aria-disabled");
  const isDisabled = ariaDisabled === "" || ariaDisabled === "true";

  return isDisabled;
}
```

```
function onKeyDownHandler(event) {
  if (event.key === " ") {
    // We have to disable default space behavior which is scrolling page down.
    event.preventDefault();
  }
  if (event.key === "Enter" && !isDisabled(event.target)) {
    onClickButtonHandler();
  }
}
```

```
function onKeyUpHandler(event) {
  if (event.key === " " && !isDisabled(event.target)) {
    onClickButtonHandler();
  }
}
```

```
<button>
  example
</button>
```

*“JavaScript should only do what
only JavaScript can do.”*

```
<div
  tabIndex={0}
  onClick={onClickButtonHandler}
  onKeyDown={onKeyDownHandler}
  onKeyUp={onKeyUpHandler}
  role="button"
>
  example
</div>
```

```
function isDisabled(eventTarget) {
  const ariaDisabled = eventTarget.getAttribute("aria-disabled");
  const isDisabled = ariaDisabled === "" || ariaDisabled === "true";

  return isDisabled;
}
```

```
function onKeyDownHandler(event) {
  if (event.key === " ") {
    // We have to disable default s
    event.preventDefault();
  }
  if (event.key === "Enter" && !isDisabled(event.target)) {
    onClickButtonHandler();
  }
}
```

```
function onKeyUpHandler(event) {
  if (event.key === " " && !isDisabled(event.target)) {
    onClickButtonHandler();
  }
}
```



button

control

imperative

control

imperative

control

declarative

<button>

example

</button>


```
button {  
  font-size: 16px;  
  padding-left: 16px;  
}
```

```
button {  
  font-size: 1rem;  
  padding-left: 1rem;  
}
```

```
button {  
  font-size: 1rem;  
  padding-inline-start: 1rem;  
}
```

```
button {  
  font-size: calc(0.5rem + 0.666vw);  
  padding-inline-start: 1em;  
}
```

```
button {  
  font-size: clamp(  
    1rem,  
    0.5rem + 0.666vw,  
    1.5rem  
  );  
  padding-inline-start: 1em;  
}
```

```
body {  
  font-size: clamp(  
    1rem,  
    0.5rem + 0.666vw,  
    1.5rem  
  );  
}
```



Supporting logical properties

September 30th, 2022

I wrote recently about [making the switch to logical properties](#) over on [The Session](#).

Initially I tried ripping the band-aid off and swapping out all the directional properties for logical properties. After all, [support for logical properties is green across the board](#).

But then I got some reports of people seeing formatting issues. These people were using Safari on devices that could no longer update their operating system. Because [versions of Safari are tied to versions of the operating system](#), there was nothing they could do other than switch to using a different browser.

I've said it before and I'll say it again, but as long as this situation continues, [Safari is not an evergreen browser](#). (I also understand that problem lies with the OS architecture—it must be incredibly frustrating for the folks working on WebKit and/or Safari.)

So I needed to add fallbacks for [older browsers that don't support logical properties](#). Or, to put it another way, I needed to add logical properties as a progressive enhancement.

"No problem!" I thought. "The way that CSS works, I can just put the logical version right after the directional version."

```
element {
  margin-left: 1em;
  margin-inline-start: 1em;
}
```

But that's not true in this case. I'm not over-riding a *value*, I'm setting two different properties.

About this site
Adactio is the online home of [Jeremy Keith](#), a web developer and author living and working in Brighton, England.
[Get in touch](#)

Customise
Choose a theme...
default ▾ ?
This is the plain vanilla look.

Search
Search the journal:

▪ Peruse the [archive](#)
▪ Browse the [tags](#)

Subscribe
You can subscribe to the [journal RSS feed](#) or you can follow [@adactioJournal](#) on Twitter.

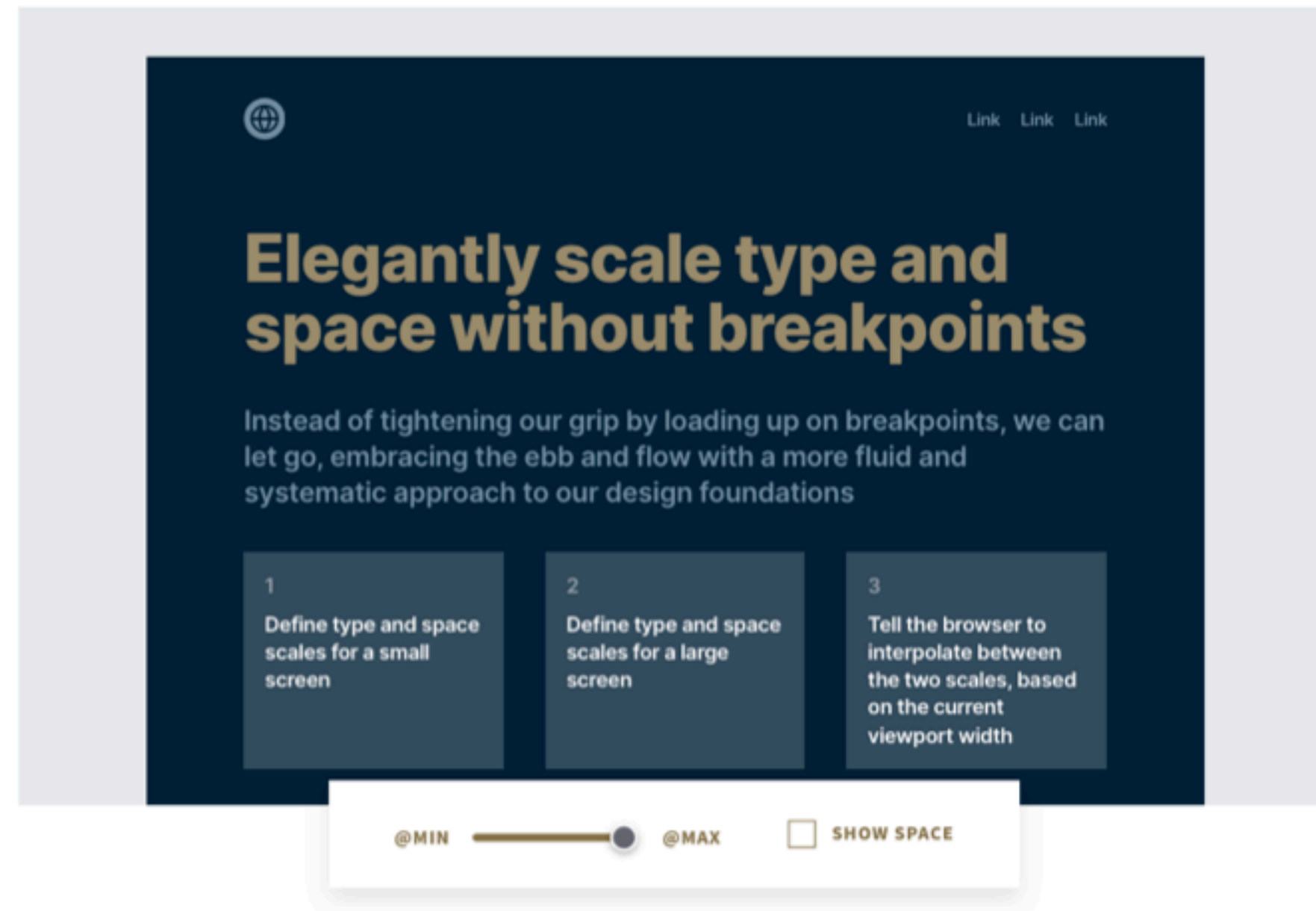
Recommended reading
Hand-picked highlights from the archive.
▪ [Split](#)



utopia.fyi

UTOPIA.
FLUID RESPONSIVE DESIGN

[Type calculator](#) [Space calculator](#) [Blog](#) [Showcase](#)



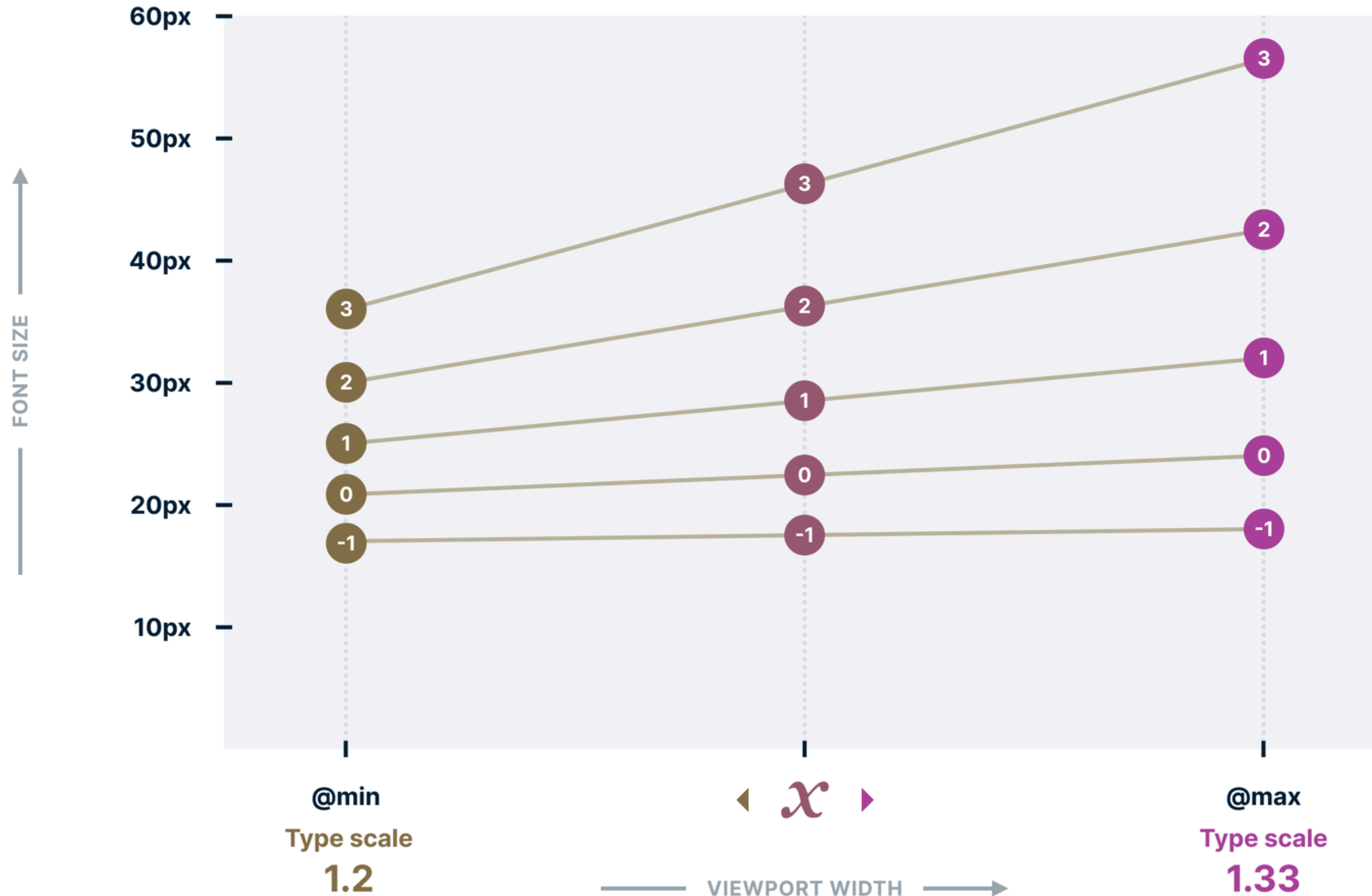
Utopia emerges when designers and developers share a systematic approach to fluidity in responsive design. Instead of designing for x number of arbitrary breakpoints, we can design a system within which elements scale proportionally and fluidly. This can help you to:

- Design and code minimally and elegantly
- Streamline collaboration between design and development roles
- Ensure visual harmony and consistency

Utopia is not a product, a plugin, or a framework. It's a memorable/pretentious word we use to refer to a way of thinking about fluid responsive design. There's no program or dependency to install, although we are developing some free tools to support your next Utopian project:



Fluid interpolation between two typographic scales by viewport width



I design these



This is a large heading lorem ipsum dolor

This text is one step up the scale from body size. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.

This is body copy lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

This is an inline heading with top and bottom spacing

This is body copy lorem ipsum dolor sit amet, consectetur adipiscing elit,

@min

Type scale

1.2

This is a large heading lorem ipsum dolor

This text is one step up the scale from body size. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.

This is body copy lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

This is an inline heading with top and bottom spacing

This is body copy dui aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

This is body copy lorem ipsum dolor sit amet, consectetur adipiscing elit,

x

Type scale



maths designs this

This is a large heading lorem ipsum dolor

This text is one step up the scale from body size. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore.

This is body copy lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

This is an inline heading with top and bottom spacing

This is body copy dui aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

@max

Type scale

1.33

calc()

clamp()

min() and **max()**

fit-content

min-content and **max-content**

repeat()

minmax()

youtube.com/layoutland

The screenshot shows a YouTube video player interface. At the top, the YouTube logo and search bar are visible. The video title is "Designing Intrinsic Layouts" by Jen Simmons, part of the "An Event Apart" series. The video has 16,564 views and was uploaded on Mar 10, 2020. The video player shows a progress bar at 0:01 / 59:07. Below the player, there are interaction buttons for LIKE, DISLIKE, SHARE, CLIP, and SAVE. The channel name "An Event Apart" is displayed with 4.01K subscribers and a red SUBSCRIBE button. A description follows: "What do CSS features like CSS Grid, Flexbox, Multicolumn, Flow layout and Writing Modes mean for our design medium? How might we better leverage the art of graphic design? How will we create something practical, usable, and realistically doable?". There are 16 comments, and the first one is from "sheriffderek" 6 months ago. A list of related videos is shown on the right, including "Atomic Design" by Brad Frost, "Designing with Grid" by Jen Simmons, and "Redefining the Technical Possibilities of CSS" by Rachel...



every-layout.dev



Learn to write better, resilient CSS

If you find yourself wrestling with CSS layout, it's likely you're making decisions for browsers they should be making themselves.

Through a series of simple, composable layouts, Every Layout will teach you how to better harness the built-in algorithms that power browsers and CSS.

[Buy Every Layout For \\$69](#)

[Read the free rudiments and axioms](#)

Already purchased Every Layout, but lost your access?

No worries. Add the email address that you used to purchase Every Layout and we'll re-send your access link.

Your email

[Get my access link](#)



builddexcellentwebsite.es

Be the browser's mentor, not its micromanager.

Give the browser some solid rules and hints, then let it make the right decisions for the people that visit it, based on their device, connection quality and capabilities. This is how they will get a genuinely great user experience, rather than a fragmented, broken one.

Key Foundations & Principles

Modern CSS with Methodologies

Instead of brute-forcing your designs together with a CSS framework, consider opting for a CSS methodology like [CUBE CSS](#), [SMACSS](#) or [BEM](#) that empowers you to write flexible, portable CSS, rather than rigid,

Fluid type & Space

Creating type scales that respond to the viewport, rather than setting explicit values for typography and space allows you to set rules once and forget about them, knowing that whatever device, regardless of its available size will be presented with appropriate sizes



declarative design

UTOPIA.
FLUID RESPONSIVE DESIGN

[Type calculator](#) [Space calculator](#) [Blog](#) [Showcase](#)

Elegantly scale type and space without breakpoints

Instead of tightening our grip by loading up on breakpoints, we can let go, embracing the ebb and flow with a more fluid and systematic approach to our design foundations

YouTube

Search

An Event Apart

JEN SIMMONS

Designing Intrinsic Layouts

Be the browser's mentor, not its micromanager.

Give the browser some solid rules and hints, then let it make the right decisions for the people that visit it, based on their device, connection quality and capabilities. This is how they will get a genuinely great user experience, rather than a fragmented, broken one.



Relearn CSS layout
by example

**3rd
EDITION**
NOW WITH
LOGICAL PROPERTIES

Learn to write better, resilient CSS

If you find yourself wrestling with CSS layout, it's likely you're making decisions for browsers they should be making themselves.

Through a series of simple, composable layouts, Every Layout will teach you how to better harness the built-in algorithms that power browsers and CSS.

Buy Every Layout For \$69

declarative > imperative?

declarative > imperative?

it depends

culture

management

imperative

management

imperative

management

declarative

culture

design systems

imperative

design systems

declarative

thinking

analytic
thinking

systems

thinking

design systems

<button>

example

</button>

A brown, rounded rectangular button with a slight gradient and a thin dark border.

BUTTON

A red, rounded rectangular button with a slight gradient and a thin dark border.

BUTTON

A yellow, rounded rectangular button with a slight gradient and a thin dark border.

BUTTON

#B45B39

#912D0E

#571D0B

#421709

#CD262F

#A60003

#6B0002

#520001

#CEA32E

#AB7906

#866004

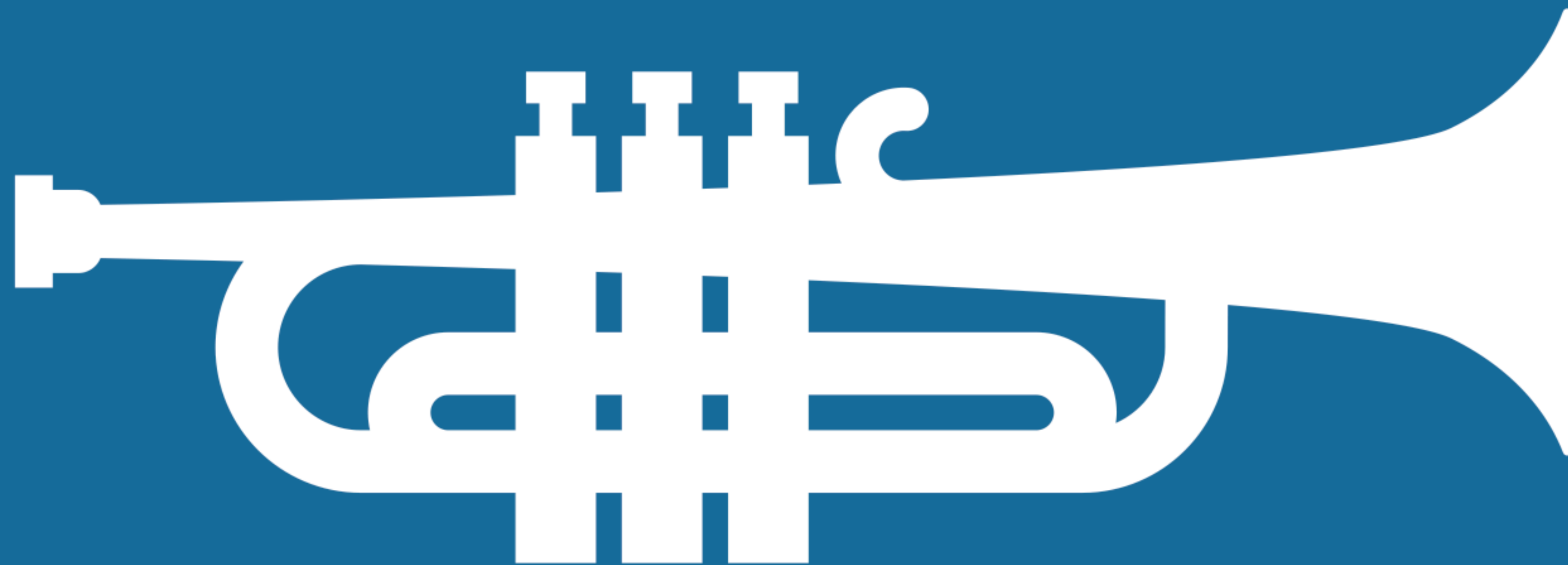
#523C02

“The border should be 10% lighter than the background color.”

declarative > imperative?

design systems

it depends



process

cloudfour.com/thinks



What We Do

Our Approach

Our Work

Articles

Speaking

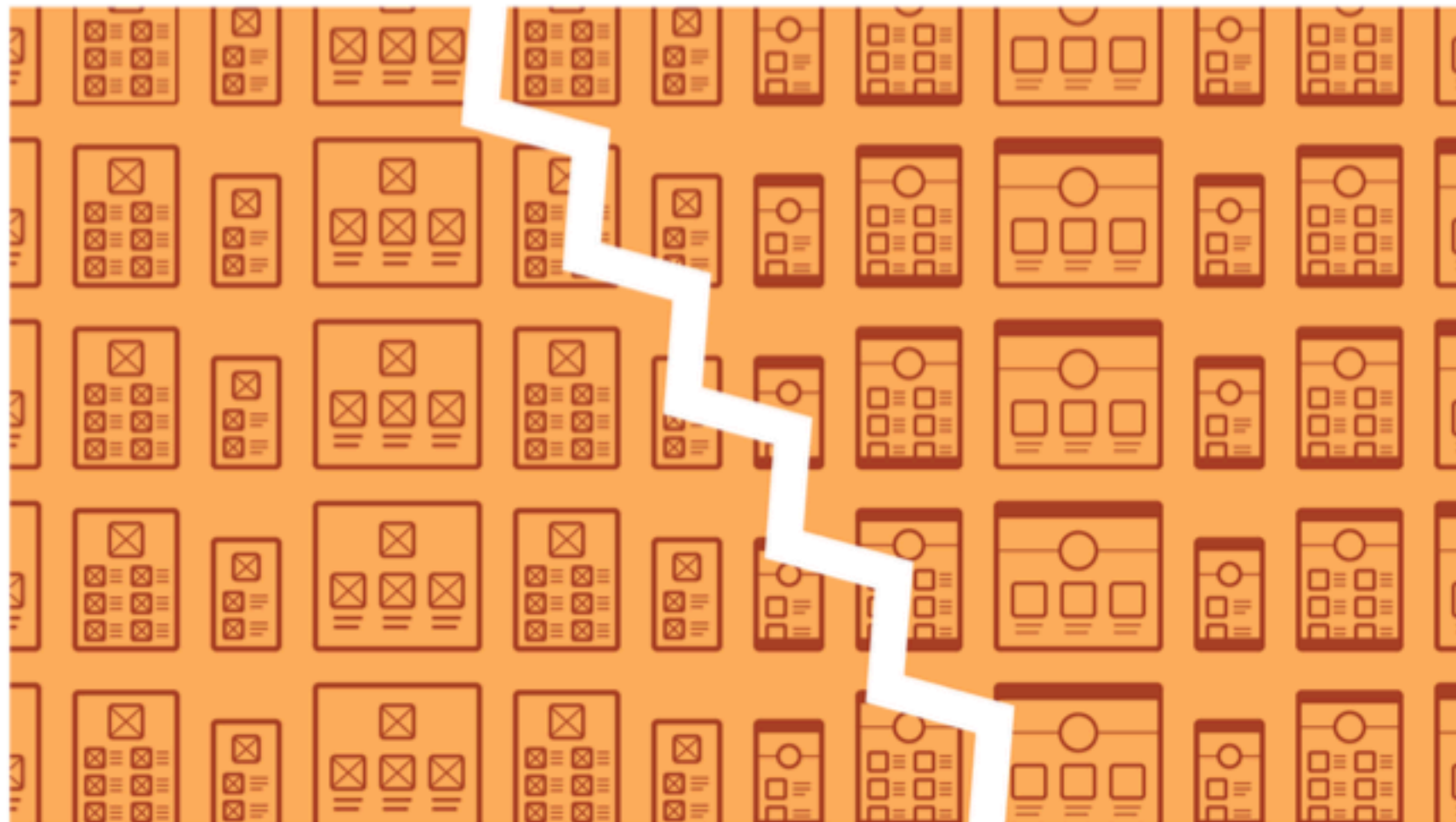
Team

Traditional Web Design Process is Fundamentally Broken



Jason Grigsby

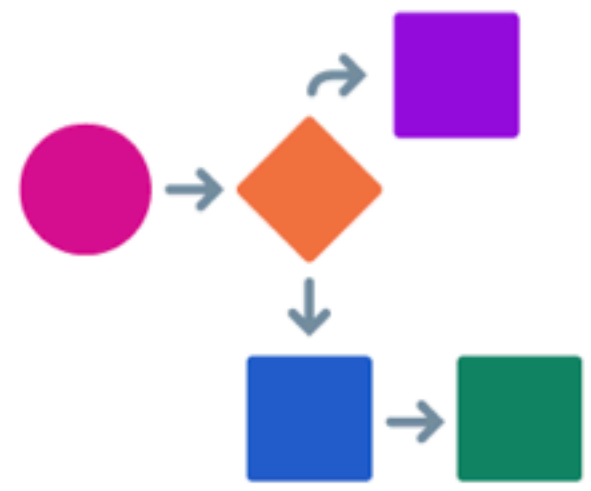
April 27th, 2022



Responsive design broke the traditional web design and development process in fundamental ways. Despite this fact, many organizations continue to use this broken process.

The traditional web design process looks something like this:





Planning



Static Wireframes



Static Mockups



Handoff



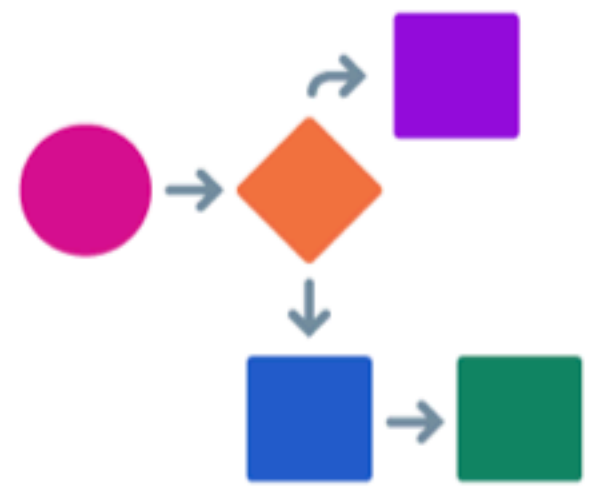
HTML, CSS and JS



Launch

Design 

Development 



Planning



Static Wireframes



Static Mockups



HTML, CSS and JS



Launch



tools

*“We shape our tools and thereafter
our tools shape us.”*

—John Culkin



CSS

“CSS is broken and I want my tools to work around the way CSS has been designed.”

“CSS is awesome and I want my tools to amplify the way that CSS has been designed.”

it depends

the medium

print

native apps

os-specific



print
native apps
os-specific



**World
Wide
Web**



World Wide Web

**World
Wide
Web**

640px

World

800px

Wide

1024px

Web

control



“The more you tighten your grip, the more the World Wide Web slips through your fingers.”

a dao of web design

a dao of web design

[ARTICLES](#)[EVENTS](#)[TOPICS](#)[WRITE FOR US](#)[LANGUAGE \(ENGLISH\) ▾](#)

A LIST APART

A Dao of Web Design

by [John Allsopp](#) · April 07, 2000

Published in [Accessibility](#), [CSS](#), [Layout & Grids](#), [Typography & Web Fonts](#)

What Zen was to the 70's (most famously with motorcycle maintenance), the *Tao Te Ching* was to the 90's. From Piglet and Pooh to Physics and back, many have sought sense in applying the *Tao Te Ching* to something (the *Tao of Physics*), or something to the *Tao Te Ching* (the *Tao of Pooh*). It can be a cheap trick, but lately it has struck me that there is more than a little to be understood about web design by looking through the prism of the Tao.

Daoism is a philosophy, like Buddhism, a way of living, of being in the world, which stems from a text of great antiquity, the *Tao Te Ching*, whose 81 "chapters" enigmatically sweep across human experience, but with a strong common theme, that of harmony.

For the last couple of years, for better or worse, my life has revolved more than a little around style sheets. I write software, tutorials, and guides for them; I've answered too many questions to count about them on newsgroups and via email; I've fought for their adoption with [The Web Standards Project](#). And slowly I've come to understand web design entirely differently because of them, and to see a strong association between design and the Tao.

What I sense is a real tension between the web *as we know it*, and the web *as it would be*. It's the tension between an existing medium, the printed page, and its child, the web. And it's time to really understand the relationship between the parent and the child, and to let the child go its own way in the world.

no comments yet

Become a patron

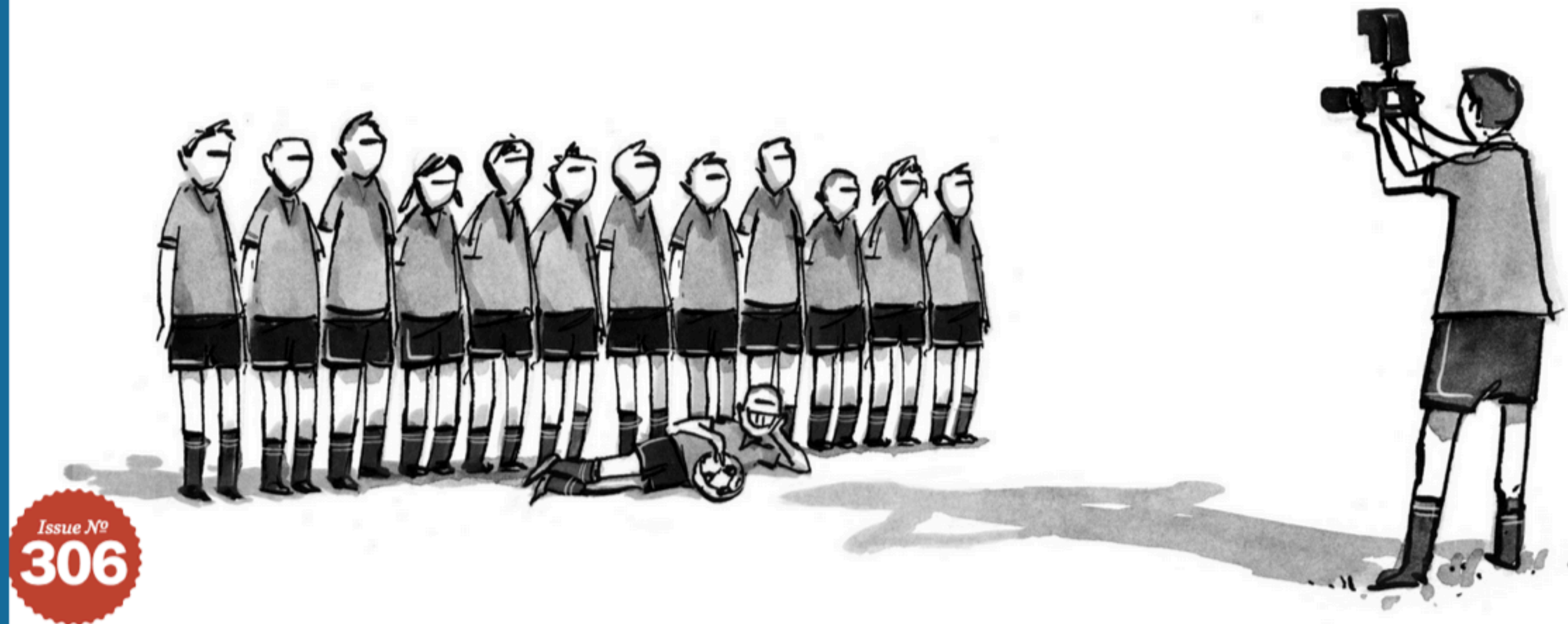
Translations

[Russian](#)

[French](#)

responsive web design

A LIST APART



Responsive Web Design

by [Ethan Marcotte](#) · May 25, 2010

Published in [CSS](#), [Interaction Design](#), [Layout & Grids](#), [Mobile/Multidevice](#), [Responsive Design](#)

“The control which designers know in the print medium, and often desire in the web medium, is simply a function of the limitation of the printed page. We should embrace the fact that the web doesn’t have the same constraints, and design for this flexibility. But first, we must ‘accept the ebb and flow of things.’”

declarative design

UTOPIA.
FLUID RESPONSIVE DESIGN

[Type calculator](#) [Space calculator](#) [Blog](#) [Showcase](#)

Elegantly scale type and space without breakpoints

Instead of tightening our grip by loading up on breakpoints, we can let go, embracing the ebb and flow with a more fluid and systematic approach to our design foundations

YouTube

Search

An Event Apart

JEN SIMMONS

Designing Intrinsic Layouts

Be the browser's mentor, not its micromanager.

Give the browser some solid rules and hints, then let it make the right decisions for the people that visit it, based on their device, connection quality and capabilities. This is how they will get a genuinely great user experience, rather than a fragmented, broken one.



Relearn CSS layout
by example

**3rd
EDITION**
NOW WITH
LOGICAL PROPERTIES

Learn to write better, resilient CSS

If you find yourself wrestling with CSS layout, it's likely you're making decisions for browsers they should be making themselves.

Through a series of simple, composable layouts, Every Layout will teach you how to better harness the built-in algorithms that power browsers and CSS.

[Buy Every Layout For \\$69](#)

Thank you