

More to give than just the div:

semantics

and how to get them right

Hidde de Vries, Access All Areas, 2021

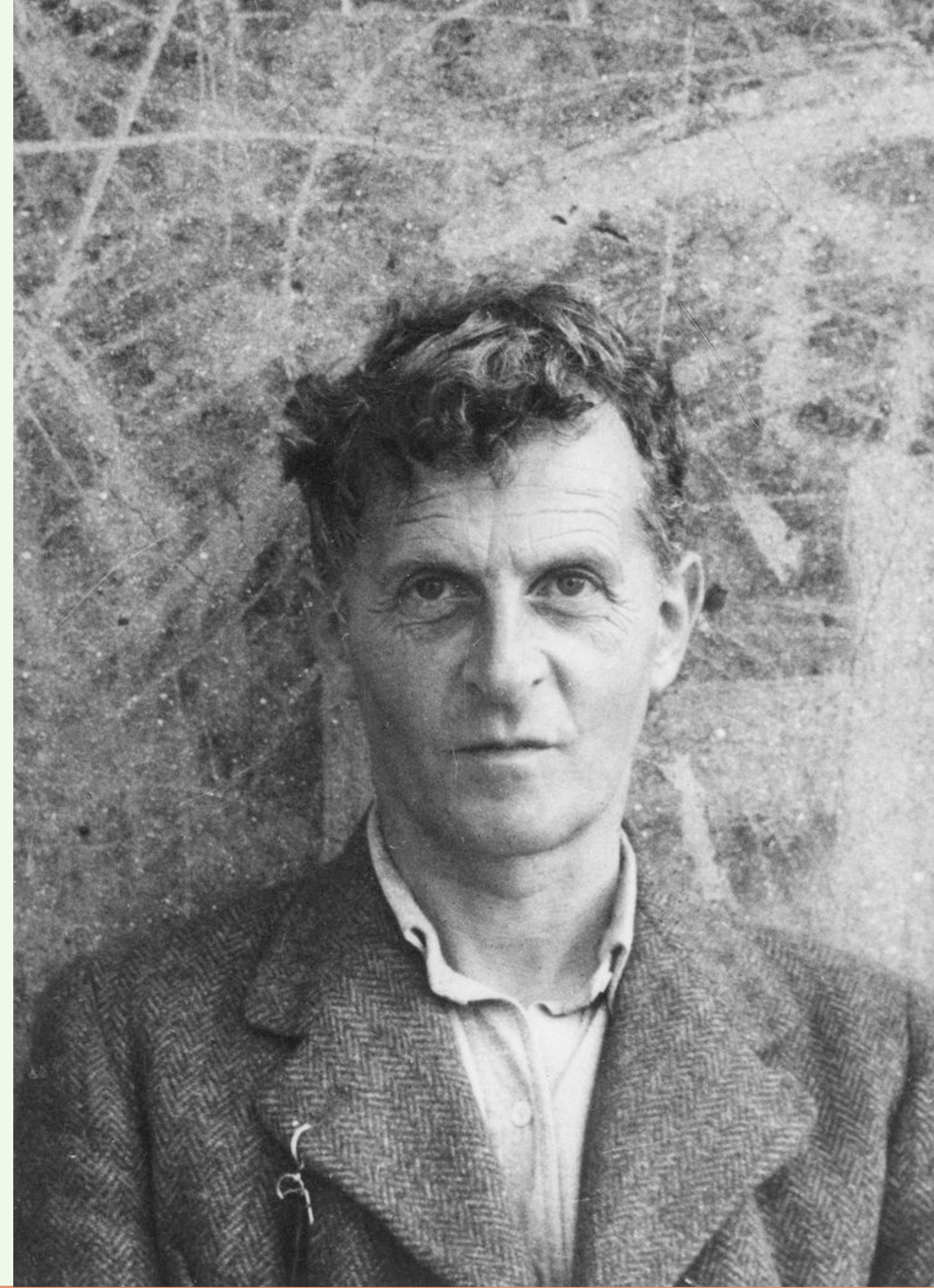
semantics



Correspondence
theories of meaning

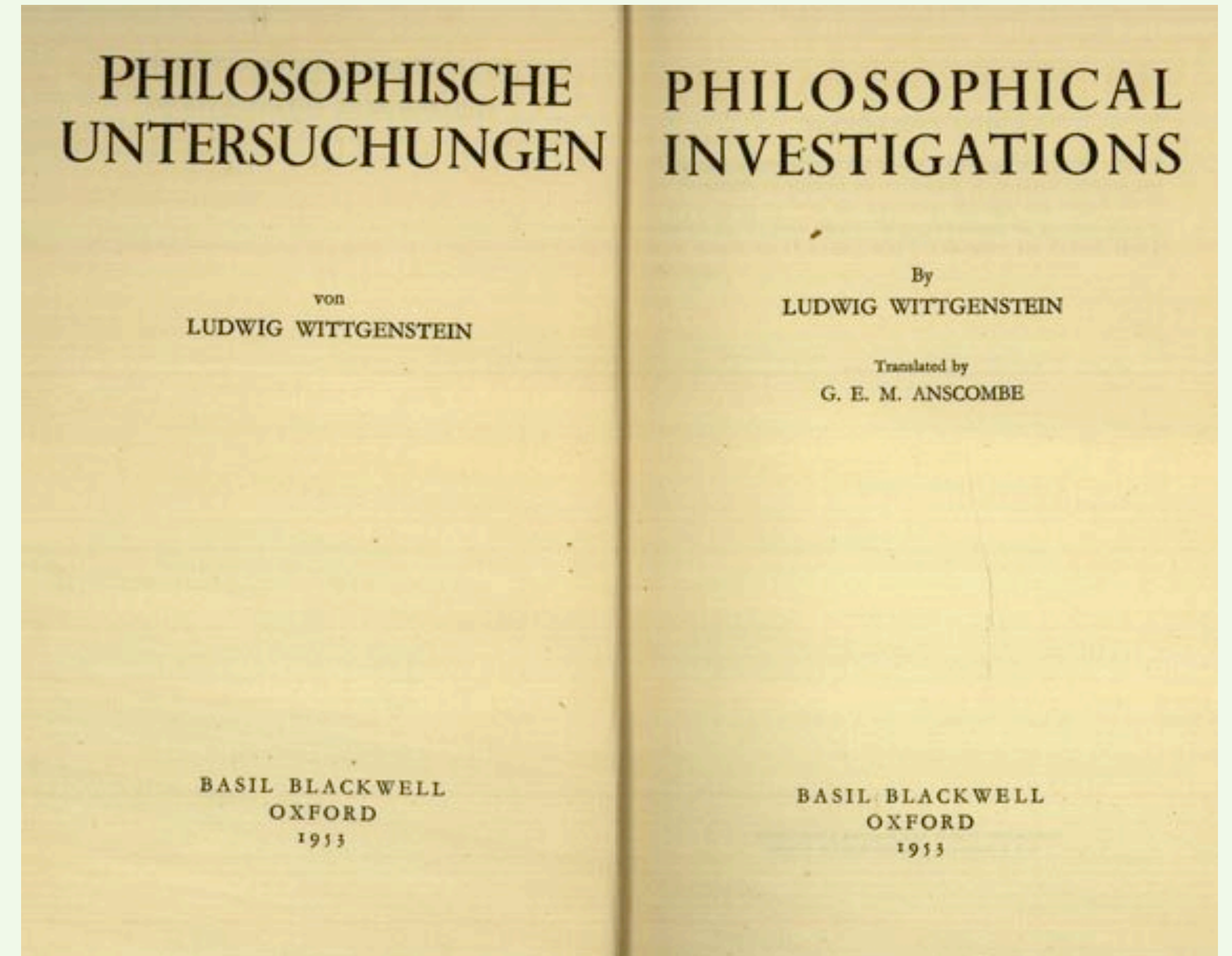
Wittgenstein: “the meaning of a word is its use in the language”

Ludwig Wittgenstein, *Philosophical Investigations* (1953)



Wittgenstein: no such thing as
a **private language**, there can
only be meaning if it's shared

Ludwig Wittgenstein, *Philosophical Investigations* (1953)

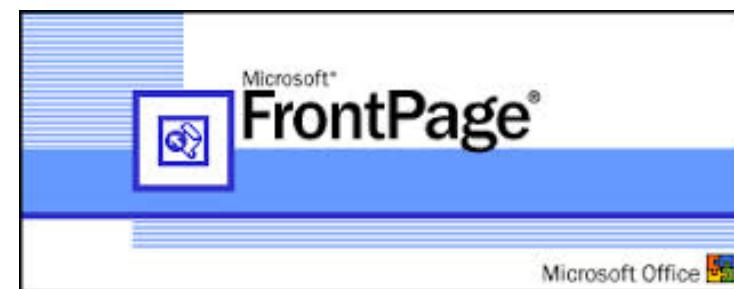
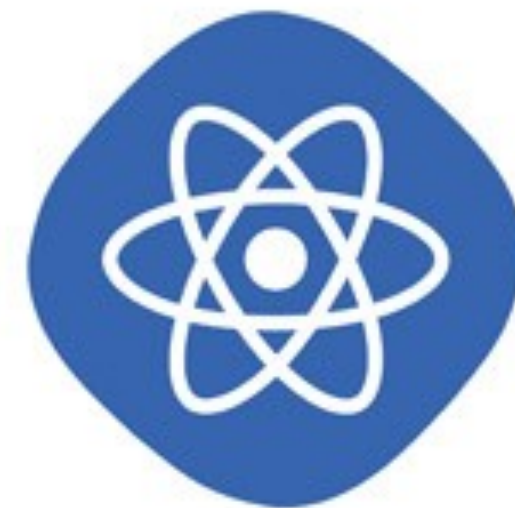


Your design system is
a shared language.



Your API is a shared language.

Your Vue/Svelte/<custom-
element> component names
are a shared language.

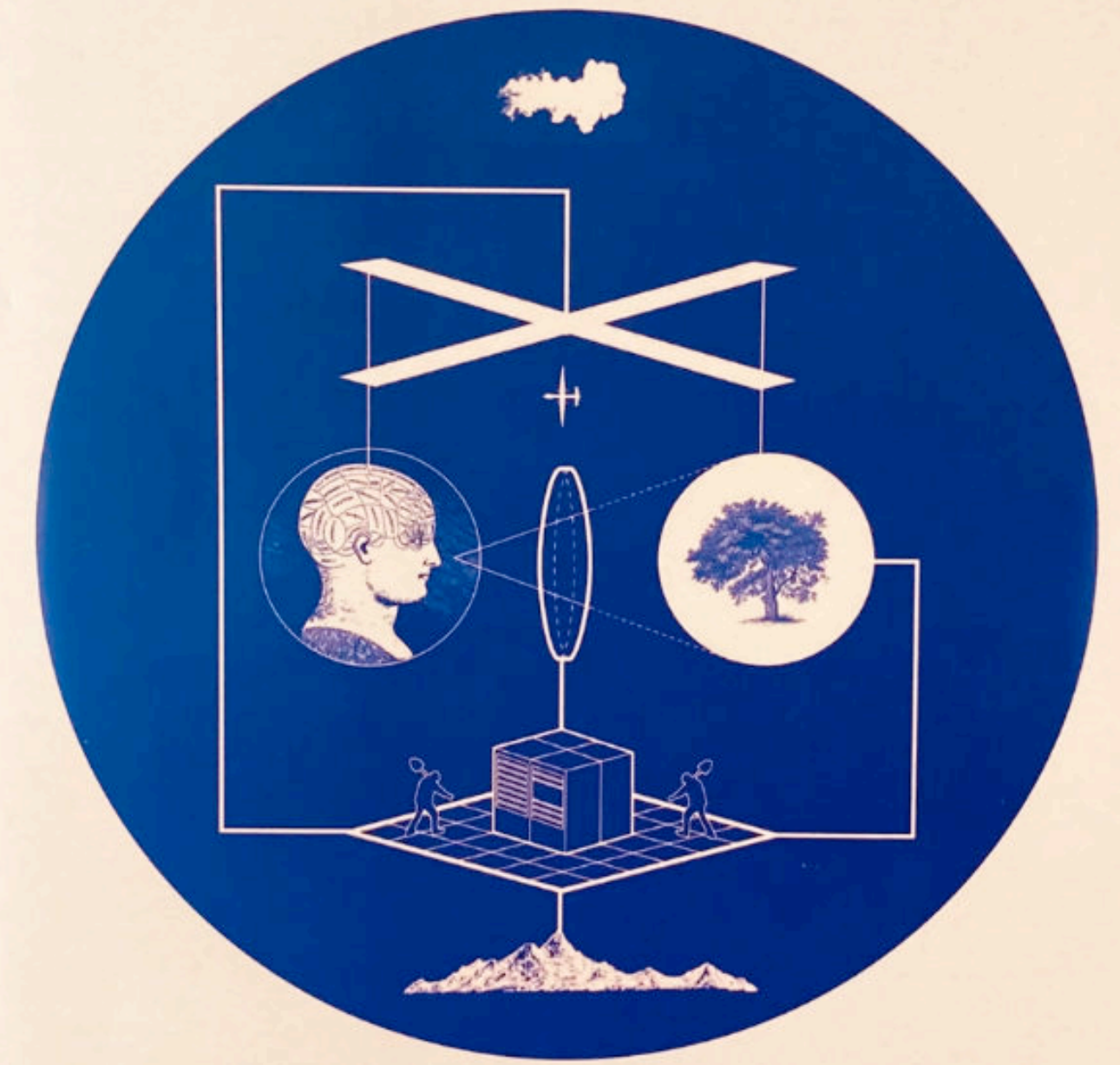


Classification depends on
place and culture



Classification is hard
and AI aren't great at it

KATE CRAWFORD



ATLAS OF AI

Semantics on the web

For the web, you could
have a lot semantics...

```
<?xml version="1.0"
<purchaseOrder orderDate="1999-10-20">
  <shipTo country="US">
    <name>Alice Smith</name>
    <street>123 Maple Street</street>
    <city>Mill Valley</city>
    <state>CA</state>
    <zip>90952</zip>
  </shipTo>
  <billTo country="US">
    <name>Robert Smith</name>
    <street>8 Oak Avenue</street>
    <city>Old Town</city>
    <state>PA</state>
    <zip>95819</zip>
  </billTo>
  <comment>Hurry, my lawn is going wild<!/comment>
  <items>
    <item partNum="872-AA">
      <productName>Lawnmower</productName>
      <quantity>1</quantity>
```



```
<xsd:element name="purchaseOrder" type="PurchaseOrderType"/>
<xsd:element name="comment" type="xsd:string"/>
<xsd:complexType name="PurchaseOrderType">
  <xsd:sequence>
    <xsd:element name="shipTo" type="USAddress"/>
    <xsd:element name="billTo" type="USAddress"/>
    <xsd:element ref="comment" minOccurs="0"/>
    <xsd:element name="items" type="Items"/>
  </xsd:sequence>
  <xsd:attribute name="orderDate" type="xsd:date"/>
</xsd:complexType>
<xsd:complexType name="USAddress">
  <xsd:sequence>
    <xsd:element name="name" type="xsd:string"/>
    <xsd:element name="street" type="xsd:string"/>
    <xsd:element name="city" type="xsd:string"/>
    <xsd:element name="state" type="xsd:string"/>
  </xsd:sequence>
</xsd:complexType>
```

...but for accessibility,
we need one,
agreed-upon set

HTML is a standard way for your website to declare its semantics

It's **not** a way to declare what
stuff looks like on a page



Multi device web

HTML enables...

**This NeXT machine was used
to develop and run the first
WWW server, multimedia
browser and web editor**





Multi device web

HTML enables...



Multi device web

Heading 1

Heading 2

Content

Default stylesheets

HTML enables...



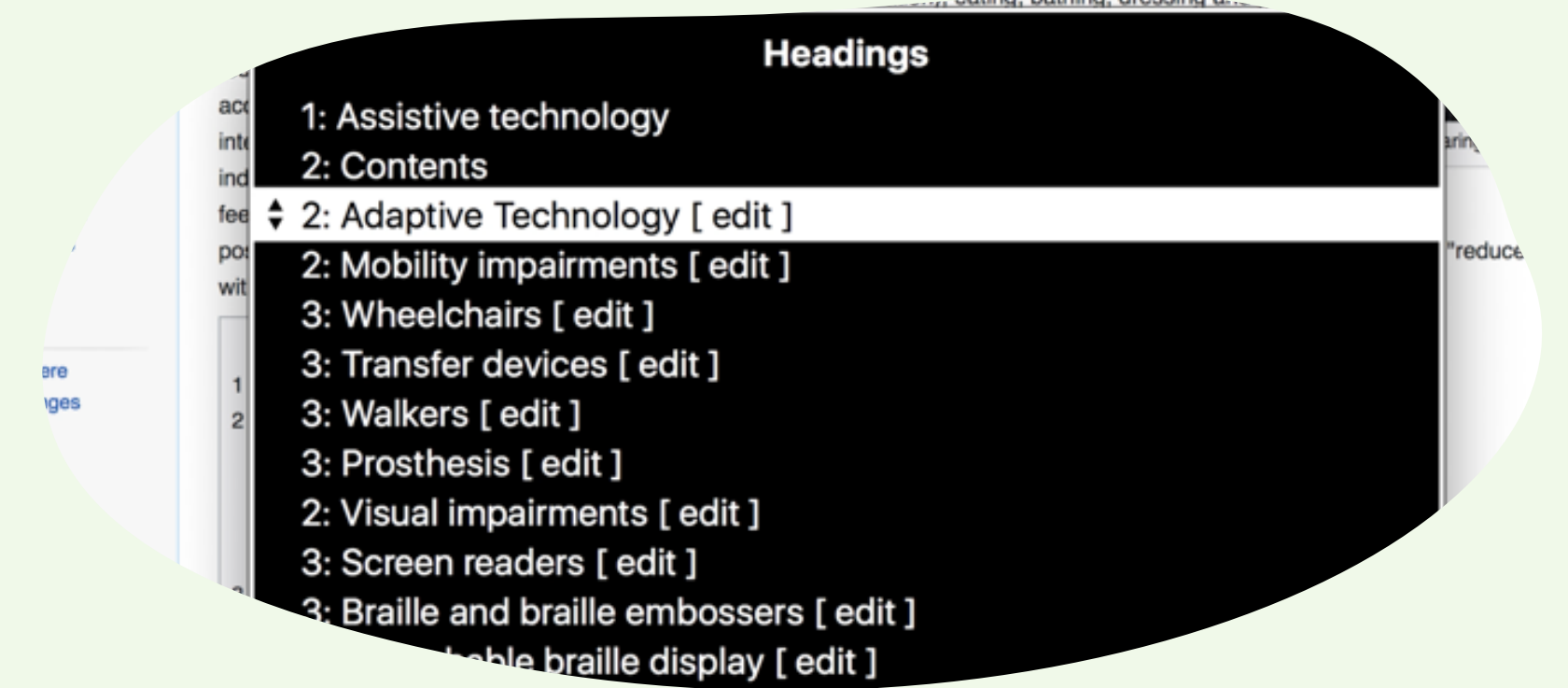
Multi device web

Heading 1

Heading 2

Content

Default stylesheets



‘Browse by heading’

HTML enables...



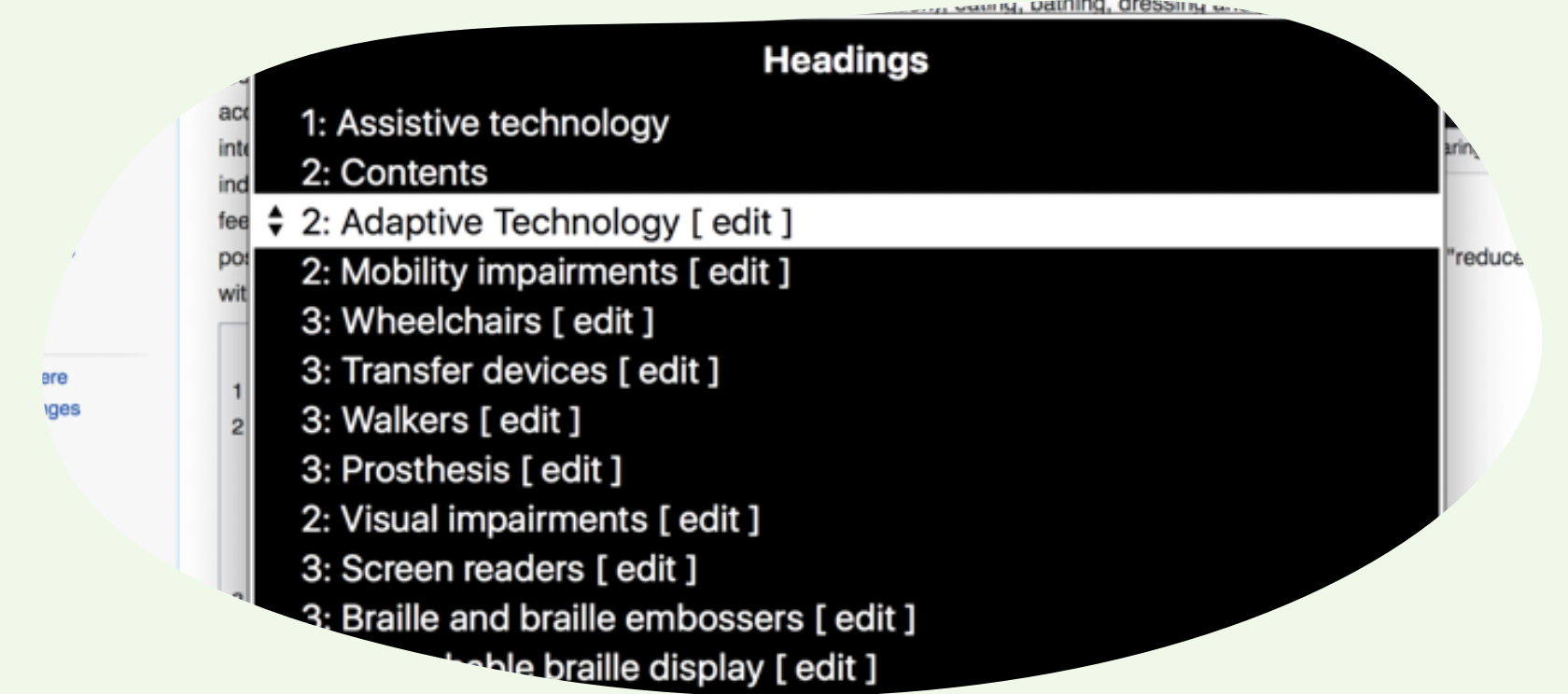
Multi device web

Heading 1

Heading 2

Content

Default stylesheets



‘Browse by heading’

HTML enables...

Default behaviour

Cook rice



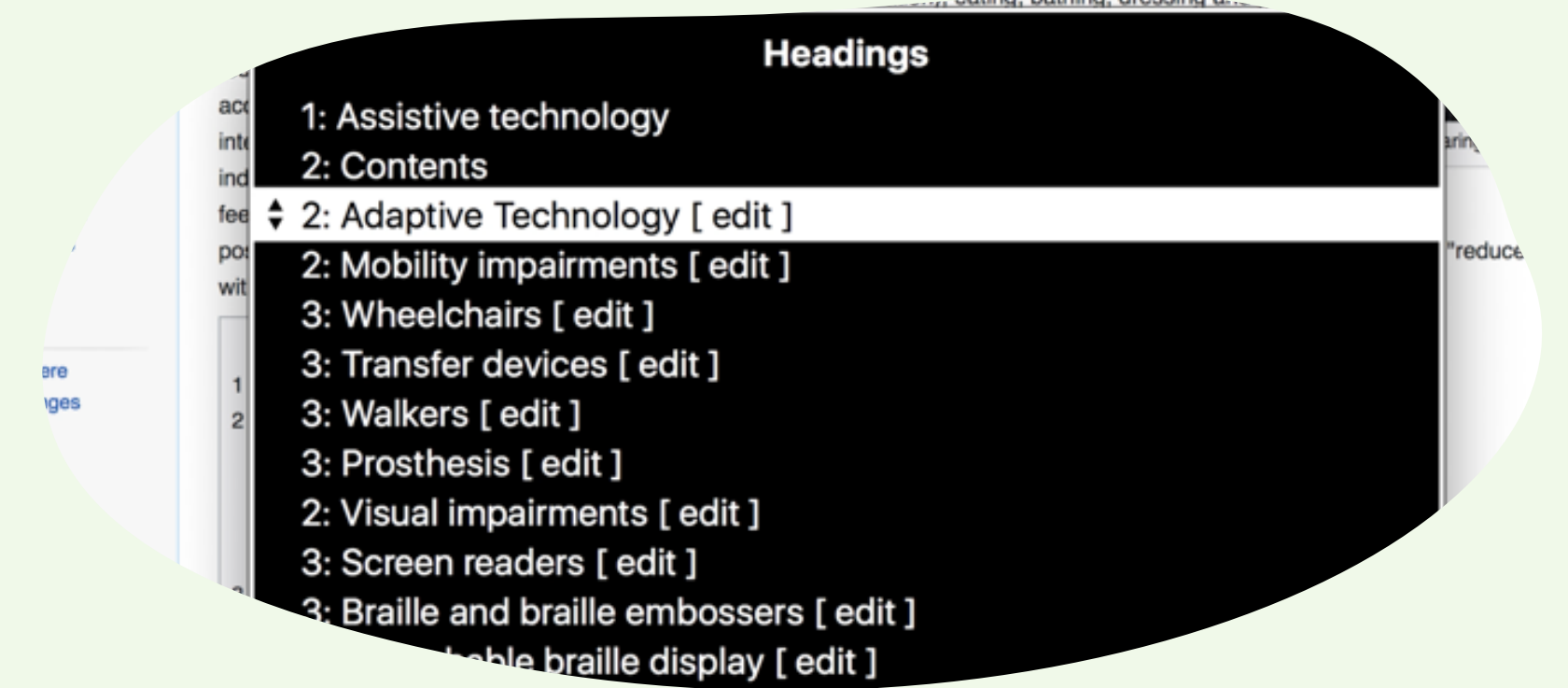
Multi device web

Heading 1

Heading 2

Content

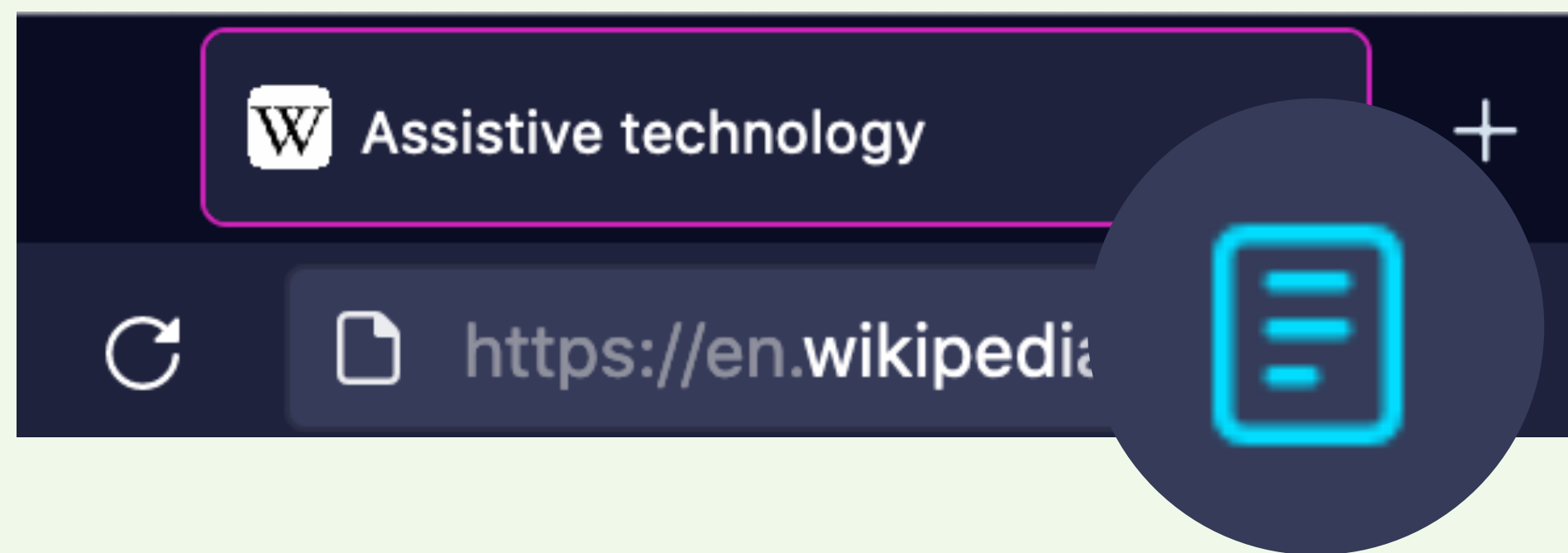
Default stylesheets



‘Browse by heading’

HTML enables...

Reader mode



Default behaviour

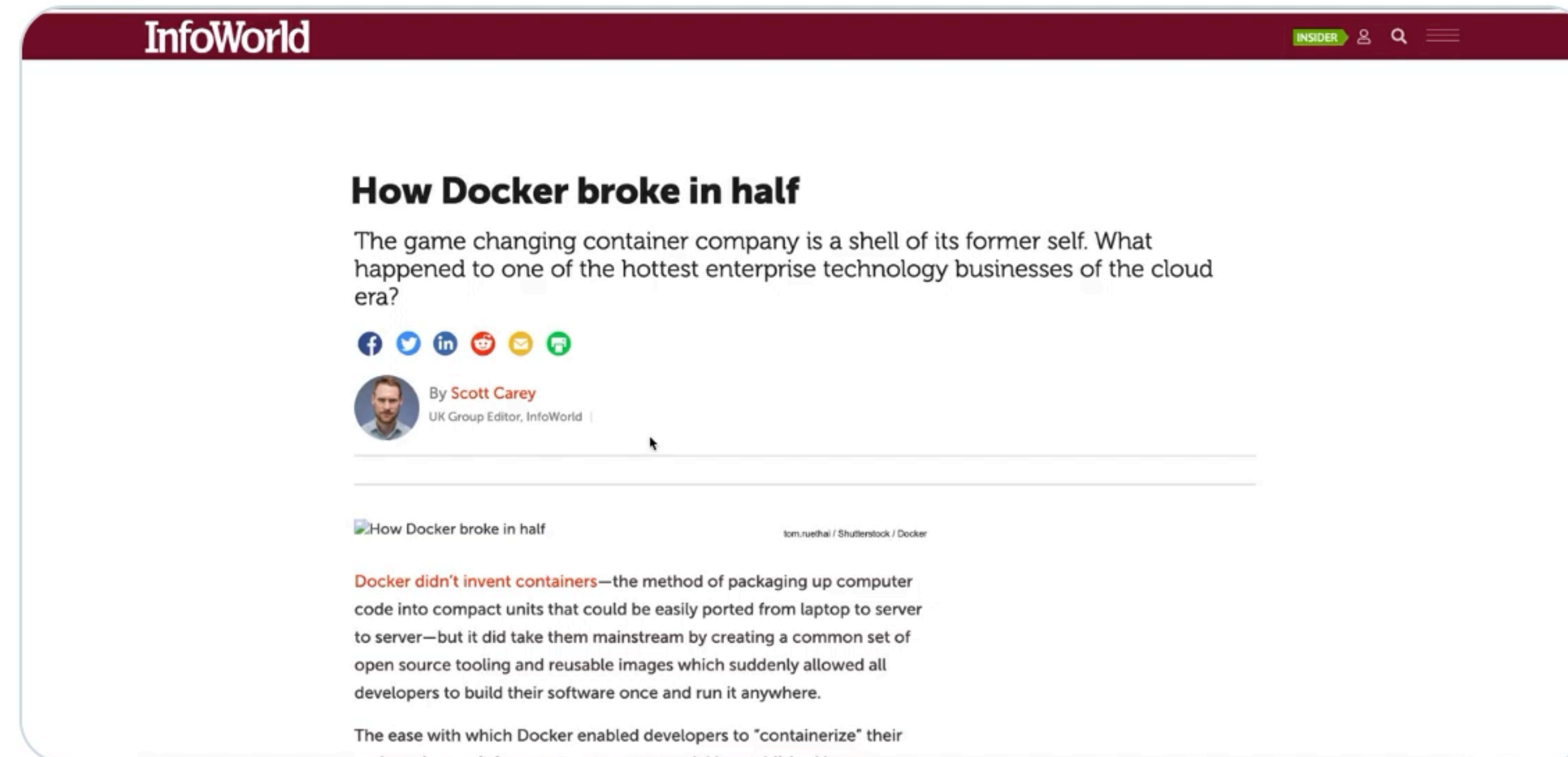
Cook rice



Andrej Karpathy ✓
@karpathy



Browsing the web, 2021



6:47 AM · Sep 9, 2021 · Twitter Web App



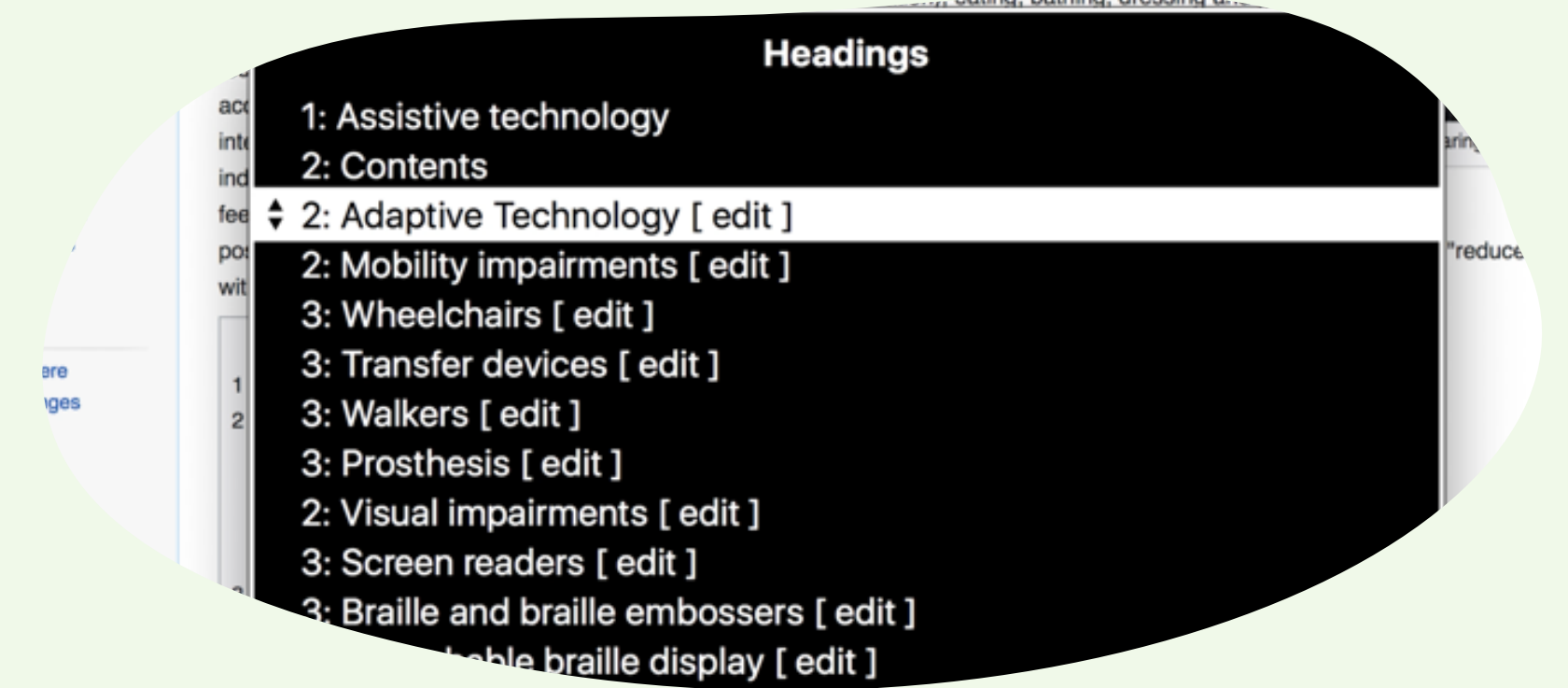
Multi device web

Heading 1

Heading 2

Content

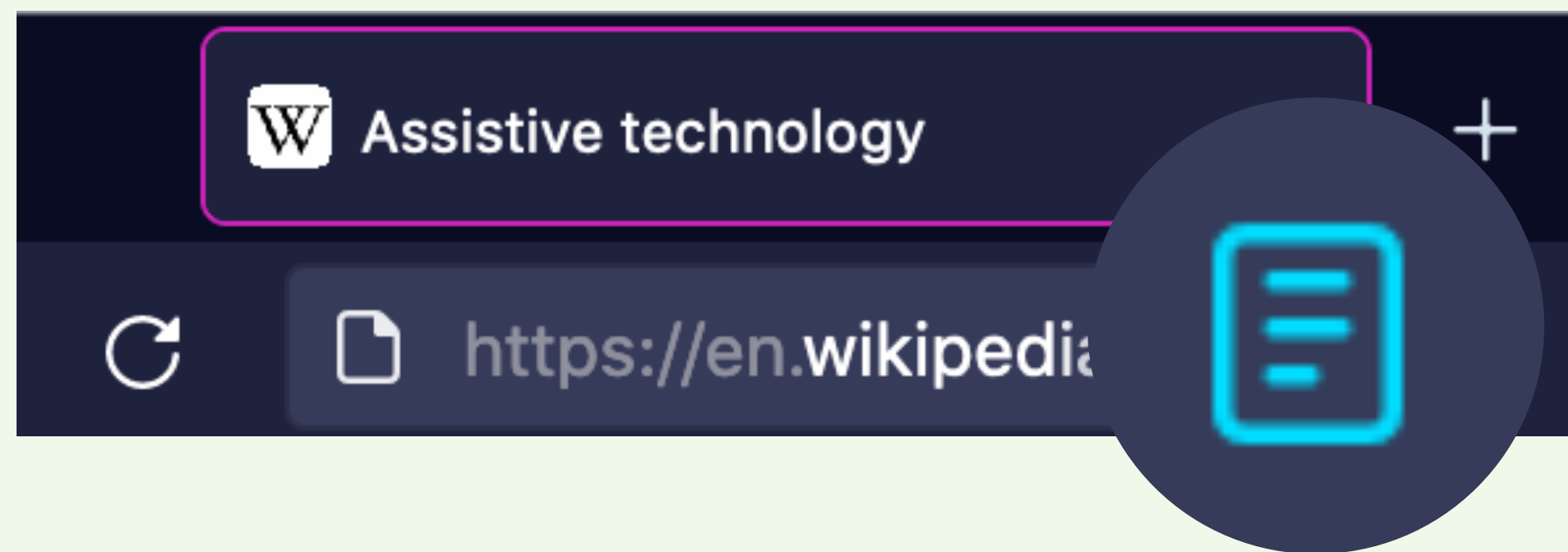
Default stylesheets



‘Browse by heading’

HTML enables...

Reader mode



Default behaviour

Cook rice



Hire for HTML expertise

In practice

Semantics in HTML is in
<elements>

Semantics in HTML is in
<elements attributes>

Semantics in HTML is in

`<elements attributes="value">`

Multiple ways to get to one semantic

`<button>`

Semantic HTML element

`<div role="button">`

Semanticless HTML element with WAI-ARIA role for semantics

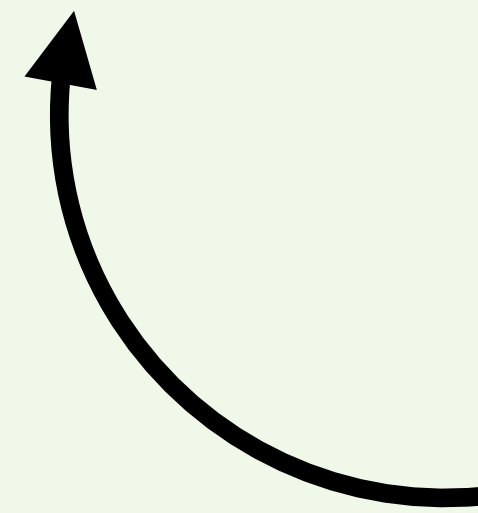
Multiple ways to get to one semantic

`<button>`

Semantic HTML element

`<div role="button">`

Semanticless HTML element with WAI-ARIA role for semantics



use if possible

Speaking of buttons...



Cory LaViska
@claviska



Unpopular opinion: in a design system, a "link" and a "button" should be one component.

10:01 PM · Oct 5, 2021 · Twitter Web App

Products

Add to cart

Products

Add to cart

the right semantics:
the right HTML
element* in the DOM

*or: the right role

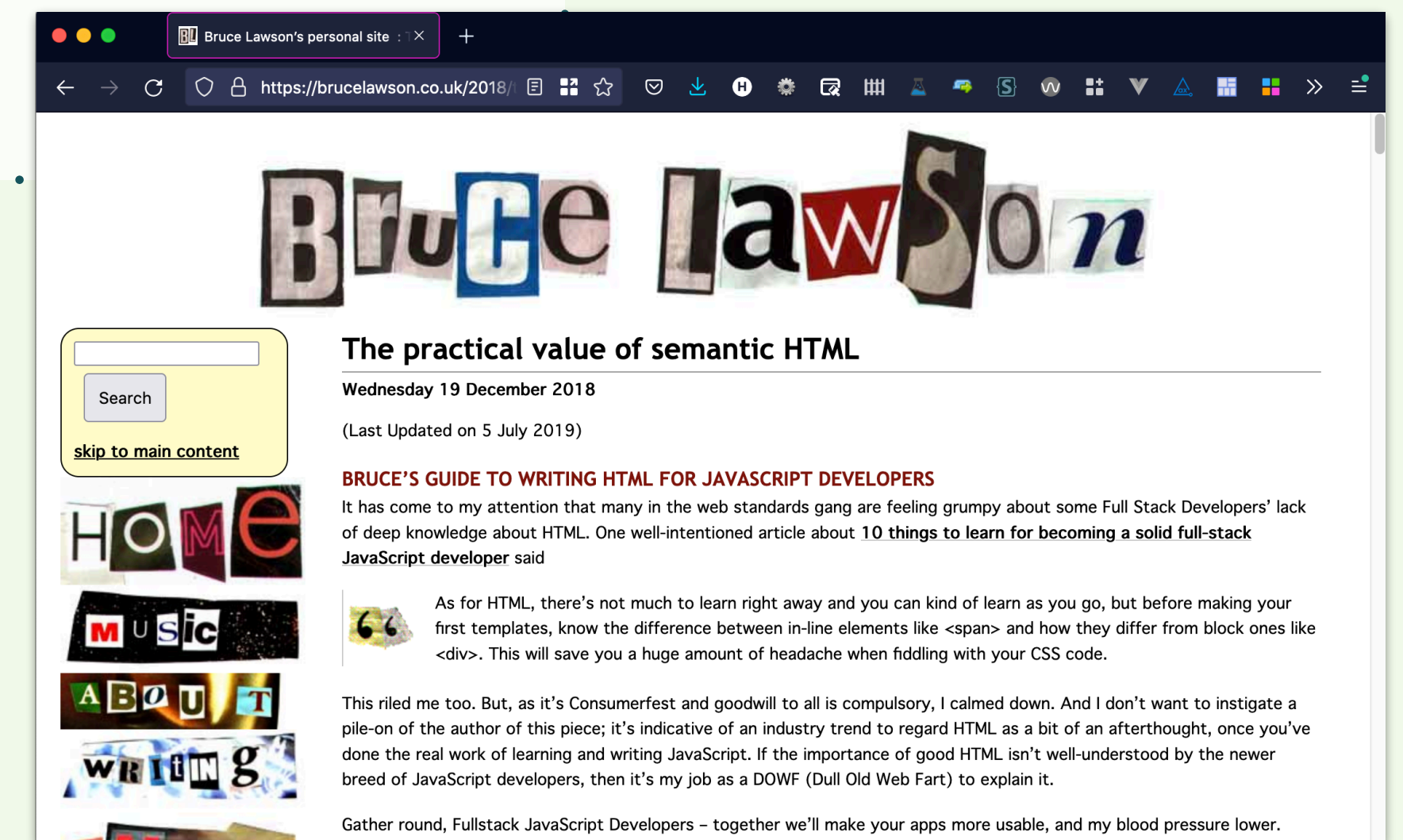
Key questions for your design system

Can it **output** the right semantics? Does it **encourage** the right semantics in docs?

meaning = use

“ (...) semantic HTML, a posh term for choosing the right HTML element for the content. This isn't a philosophical exercise; it has **directly observable practical benefits**.

Bruce Lawson, “The practical value of semantic HTML”



If you use <h1>, <h2>, <h3>..., people can:

- see headings in reader mode
- navigate by heading

The screenshot shows a Wikipedia article titled "Assistive technology". The article text describes it as an umbrella term for devices and processes used by people with disabilities. A dark overlay titled "Headings" is positioned on the right side of the article, listing the following structure:

- 1: Assistive technology
- 2: Contents
 - 2: Adaptive Technology [edit]
 - 2: Mobility impairments [edit]
 - 3: Wheelchairs [edit]
 - 3: Transfer devices [edit]
 - 3: Walkers [edit]
 - 3: Prosthesis [edit]
 - 2: Visual impairments [edit]
 - 3: Screen readers [edit]
 - 3: Braille and braille embossers [edit]
 - 3: Refreshable braille display [edit]
 - 3: Desktop video magnifier [edit]
 - 3: Screen magnification software [edit]
 - 3: Large-print and tactile keyboards [edit]
- 4 Personal emergency response systems
- 5 Accessibility software
- 6 Hearing impairments

open-borders.pages — Edited

View

Zoom

163%

Insert

Table

Chart

Text

Shape

Media

Comment

Collaborate

Format

Document

Table of Contents

Text

Select paragraph styles whose text will appear in the table of contents:

Paragraph Styles	#'s
<input type="checkbox"/> Body	<input checked="" type="checkbox"/>
<input type="checkbox"/> Blockquote	<input checked="" type="checkbox"/>
<input type="checkbox"/> Bibliography entry	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/> Header 3	
<input type="checkbox"/> Header 1	
<input checked="" type="checkbox"/> Header 2	
<input type="checkbox"/> Header & Footer	
<input type="checkbox"/> Footnote 1	

Range

Entire Document

Body

Paragraph Styles

Title

Subtitle

Heading

Heading 2

Heading 3

Heading Red

✓ Body

CAPTION

Header & Footer

Footnote

1. Introduction

Structure

The debate: closed borders versus open borders

Who wants to cross borders?

Perspective

2. The case for closed borders

The right to self-determination

The freedom of association argument

Preservation of culture

Associative ownership

Summary

3. The case for open borders

Carens' case for open borders

All humans are equal

Applicability of democratic justification

Emigration requires immigration

Multiculturalism

Summary

4. The feasibility of open borders

The political reality

4

5

5

7

7

9

9

10

12

14

17

18

18

21

22

24

25

26

28

28

If you use `<button>`, people can:

- find it in the TAB order
- press it with just a keyboard
- submit the form, even if your JavaScript failed

If you use ``, `` or `<dl>`, people can:

- Hear it's a list and with how many items
- Still see it's a list in reader mode

If you use autocomplete on inputs, users can:

- Tell their browser to fill in data for them
- Assistive tech can announce input purpose
- Use plugins for personalisation, e.g. with icons

If you use `table` with a `caption`, this for headers and `scope` attributes for direction:

- Assistive tech can provide useful affordances to end users

The HTML spec
is the place to find
out how to use HTML

developers.whatwg.org



HTML: The Living Standard

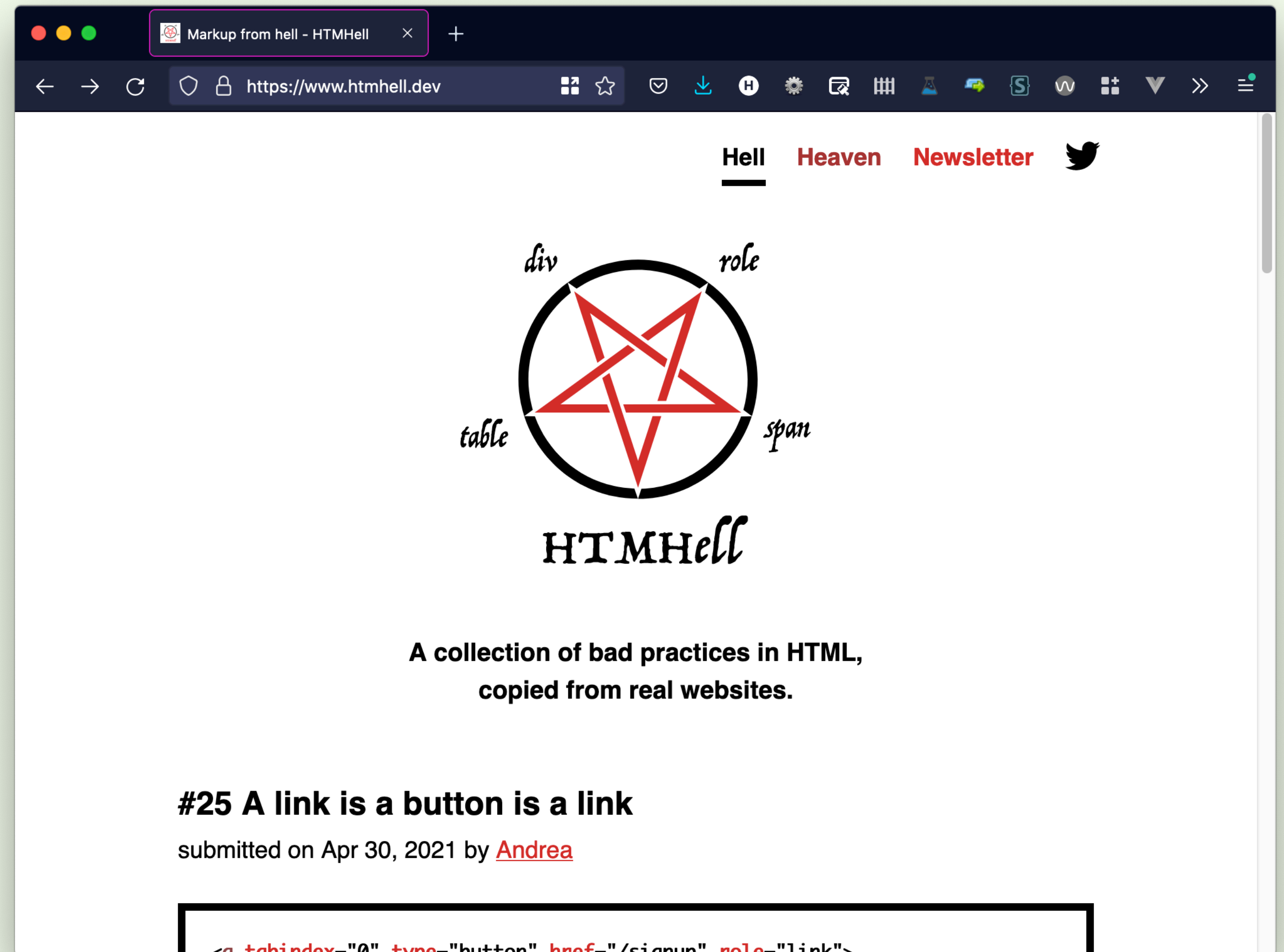
Developer's Edition — Last Updated 18 September 2017

Search. Press

TABLE OF CONTENTS

- 1 [Introduction](#)
 - 1.1 [Is this HTML5?](#)
 - 1.2 [Background](#)
 - 1.3 [Audience](#)
 - 1.4 [Scope](#)
 - 1.5 [History](#)
 - 1.6 [Design notes](#)
 - 1.7 [HTML vs XML syntax](#)
 - 1.8 [Structure of this specification](#)
 - 1.9 [Privacy concerns](#)
 - 1.10 [A quick introduction to HTML](#)
 - 1.11 [Conformance requirements for authors](#)
 - 1.12 [Suggested reading](#)

HTMHell collects examples of how **not** to use HTML



Intentions vs reality

Sometimes, semantics are
undone if you use specific CSS



display overrides semantics

```
display: block;  
display: inline;  
display: grid;  
display: flex;  
display: contents;  
display: none;
```

Safari / macOS 10.15.6

CSS	<table>, etc.	, , <dl>	<h#>	<button>
display: flex	✗ ¹	✓ ⁴	✓	✓
display: grid	✗ ¹	✓ ⁴	✓	✓
display: block	✗ ²	✗	✓	✓
display: inline-block	✗ ²	✗	✓	✓
display: contents	✗ ³	✗	✗	✗

Lists can lose list semantics when you
set `list-style-type: none` (in Safari)

- courgette
- feta
- basil
- garlic



25% KORTING



Oatly! Haverdrink Barista Edition

~~2.65~~
1.99
1 l



Valess Beefstyle burger
Advertentie

3.59
200 g



AH Amandel drink ongezoet

1.65
1 l



1.65
1 l



1.79
1 l



0.88
1 l



James Craig  (he/him)

@cookiecrook



Replying to [@pgrucza](#) [@rob_dodson](#) and 5 others

Correct. Check the test case linked in that Bugzilla tracker. Lististis on the Web was one of our biggest complaints from VoiceOver users prior to the heuristic change. FWIW, all browsers have similar heuristic checks for layout tables.

6:24 PM · Jan 12, 2019 · Twitter Web App



James Craig 🪄 (he/him) @cookiecrook · Jan 12, 2019



This decision was all about the users' experience on the majority of pages where web developers are not paying attention to the screen reader experience. Definitely open to change suggestions (including updating the heuristic) that make it better for authors w/o penalizing users.



4



1



3



Sometimes, semantics are undone
when you nest unexpectedly

Semantics in summary

```
<details>  
  <summary>Ingredients</summary>  
  <ul>  
    <li>courgette</li>  
    <li>peas</li>  
    <li>basil</li>  
    <li>feta</li>  
  </ul>  
</details>
```

Semantics in summary

```
<details>
  <summary>
    <h2>Ingredients</h2>
  </summary>
  <ul>
    <li>courgette</li>
    <li>peas</li>
    <li>basil</li>
    <li>feta</li>
  </ul>
</details>
```

Content model:
Phrasing content,
optionally intermixed with
heading content.

Heuristics in the platform and assistive technologies

When you set `text-transform: uppercase`, some assistive technologies will read it like an abbreviation

[Buy tickets](#)[Your tickets](#)[Stations and destinations](#)[Travel information](#)[Travelling with us](#)[Help and support](#)

Christmas tickets on sale now

Book early and save for a fun-filled
Christmas.

[Book now](#)

Book up to 7 January 2022 on selected routes

[Buy tickets](#)[Check your journey](#)

`::before / ::after`
is included in accessible name calculation

::before / ::after is included in accessible name calculation

```
<button>Buy product</button>
```

A teal button with the text "Buy product" in white.

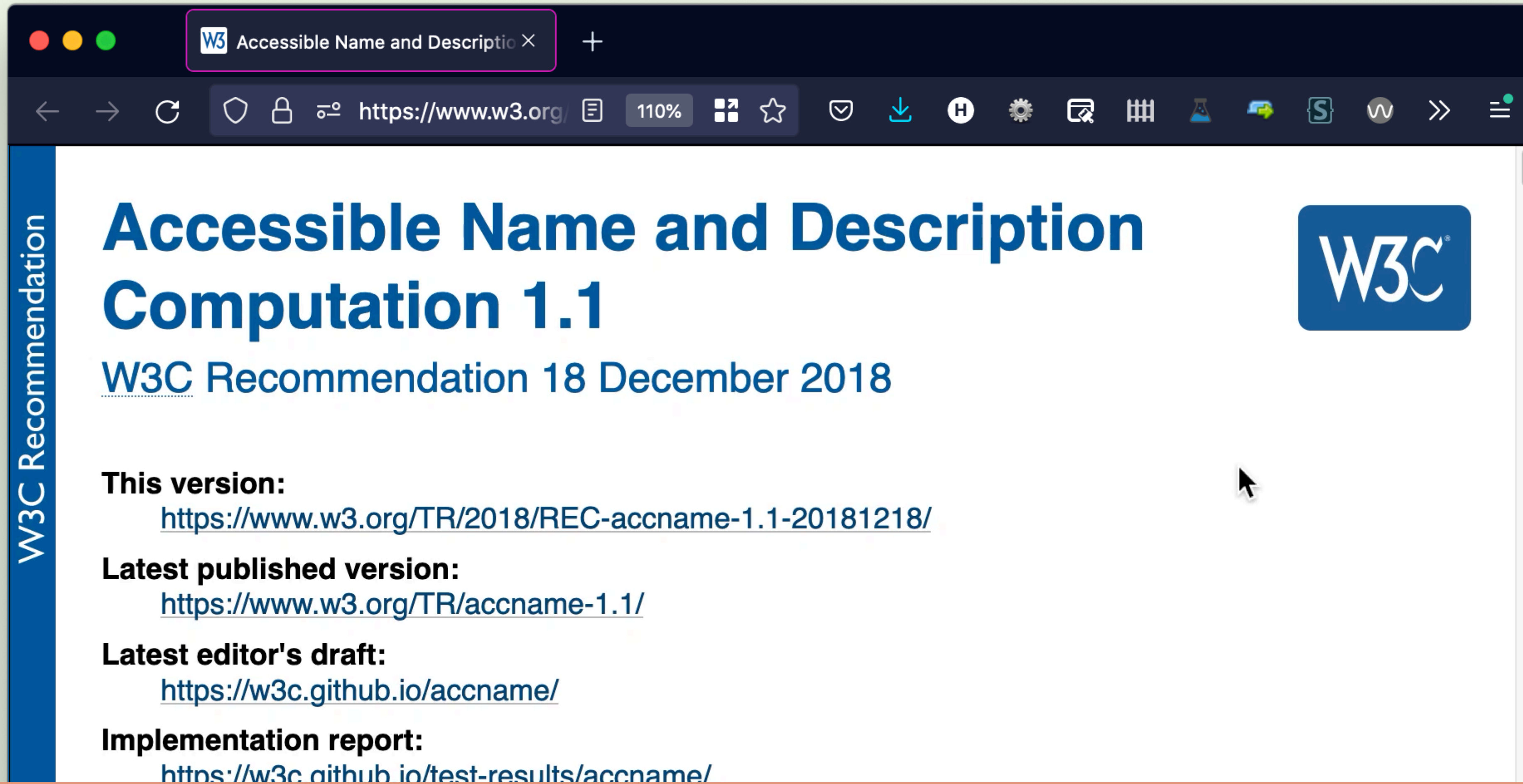
::before / ::after is included in accessible name calculation

```
<button>Buy product</button>
```

```
button::before {  
  content: "💰";  
}
```



::before / ::after is included in accessible name calculation



A screenshot of a web browser displaying the W3C page for "Accessible Name and Description Computation 1.1". The browser's address bar shows the URL "https://www.w3.org/". The page features a blue header with the title "Accessible Name and Description Computation 1.1" and the W3C logo. Below the title, it states "W3C Recommendation 18 December 2018". A vertical blue sidebar on the left contains the text "W3C Recommendation". The main content area lists four links: "This version:", "Latest published version:", "Latest editor's draft:", and "Implementation report:", each followed by a URL. A mouse cursor is visible over the page.

W3C Accessible Name and Description ×

https://www.w3.org/ 110%

Accessible Name and Description Computation 1.1

W3C Recommendation 18 December 2018

This version:
<https://www.w3.org/TR/2018/REC-accname-1.1-20181218/>

Latest published version:
<https://www.w3.org/TR/accname-1.1/>

Latest editor's draft:
<https://w3c.github.io/accname/>

Implementation report:
<https://w3c.github.io/test-results/accname/>

The future

Our design systems
commonly contain things that
are not built into HTML

Sometimes, our design systems
contain things that do exist, but not
with our desired level of **style-ability**

“We hope to make it unnecessary to
reinvent built-in UI controls

Open UI Homepage, <https://open-ui.org>



Goals of OpenUI

Document
component names
as they exist today

A common language
for describing UIs
and design systems

Browser standards
for web app
components

Goals of OpenUI

Document
component names
as they exist today

A common language
for describing UIs
and design systems

Browser standards
for web app
components

Goals of OpenUI

Document
component names
as they exist today

A common language
for describing UIs
and design systems

Browser standards
for web app
components

Could AI guess semantics?

Conclusion

Semantics only works if shared (use the HTML standard)

Semantic HTML has many benefits, some unexpected

Beware of how CSS, ARIA and AT can impact semantics



Thanks for listening!

hidde@hiddedevries.nl

hidde.blog

[Links on talks.hiddedevries.nl](https://talks.hiddedevries.nl)