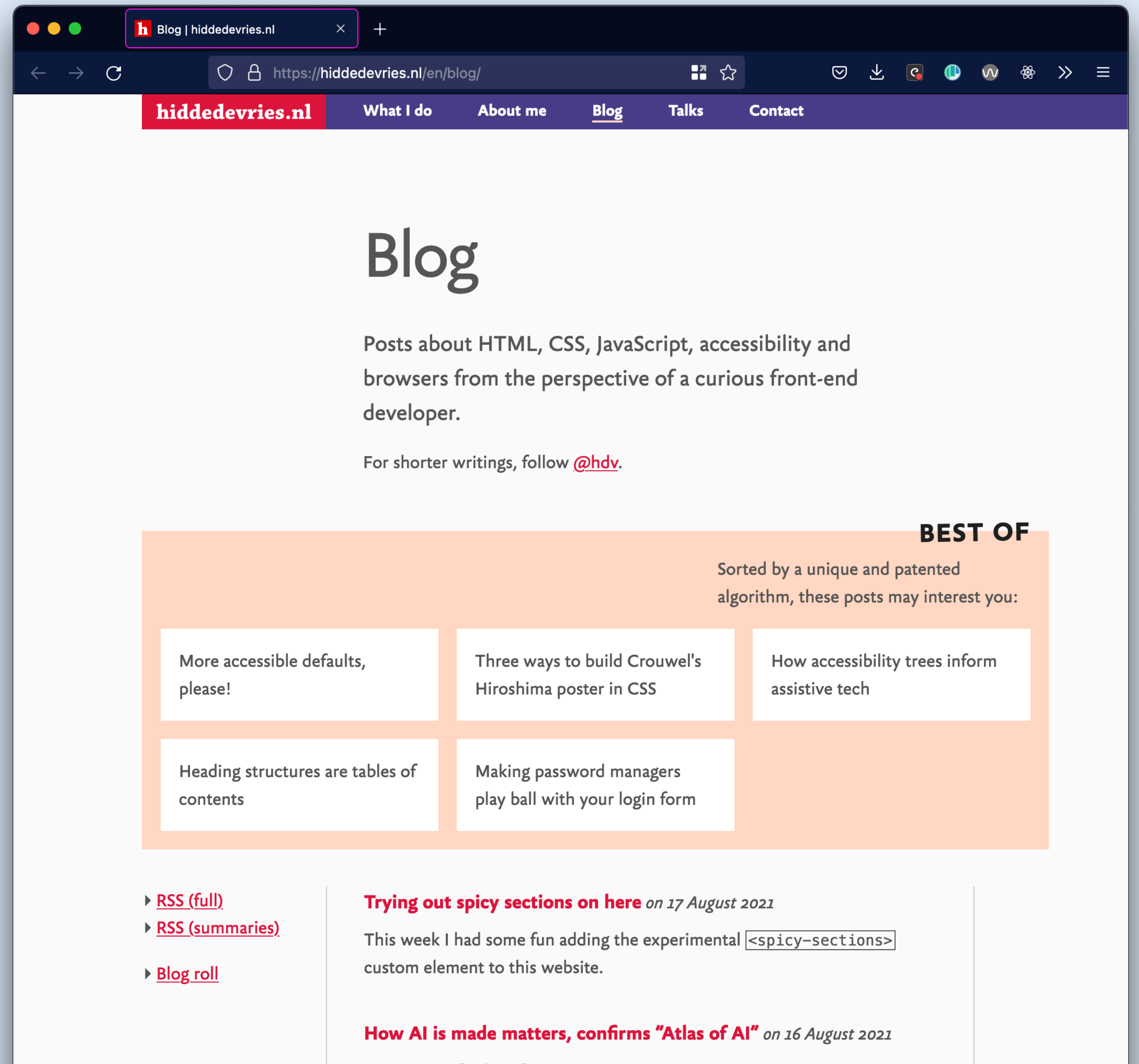


It's the
midnight

Hidde de Vries / CodeLand 2021

that
matters

hidde.blog



security

performance

Quality of websites

accessibility

usability

Accessible to

**People with
permanent
disabilities**



**People with
temporary
disabilities**

**People in
certain
situations**



Responsibility of

Content designers

Web developers

UI designers

CMS integrators

User researchers

Responsibility of

Content designers

Web developers

UI designers

CMS integrators

User researchers

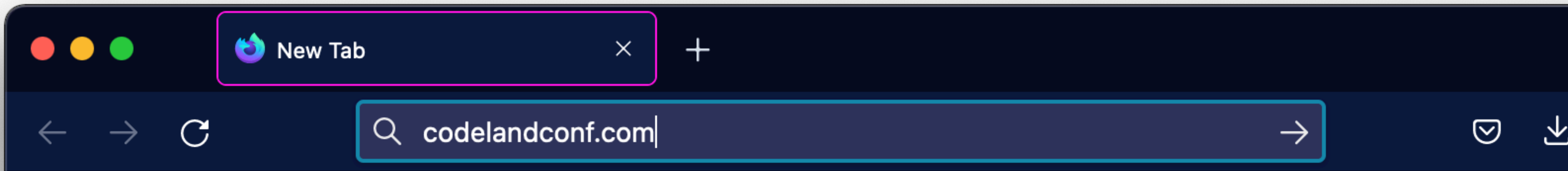
Code

Assistive Technologies (AT)



text-to-speech
screen magnifiers
alternate pointing devices







CodeLand Conference 2021



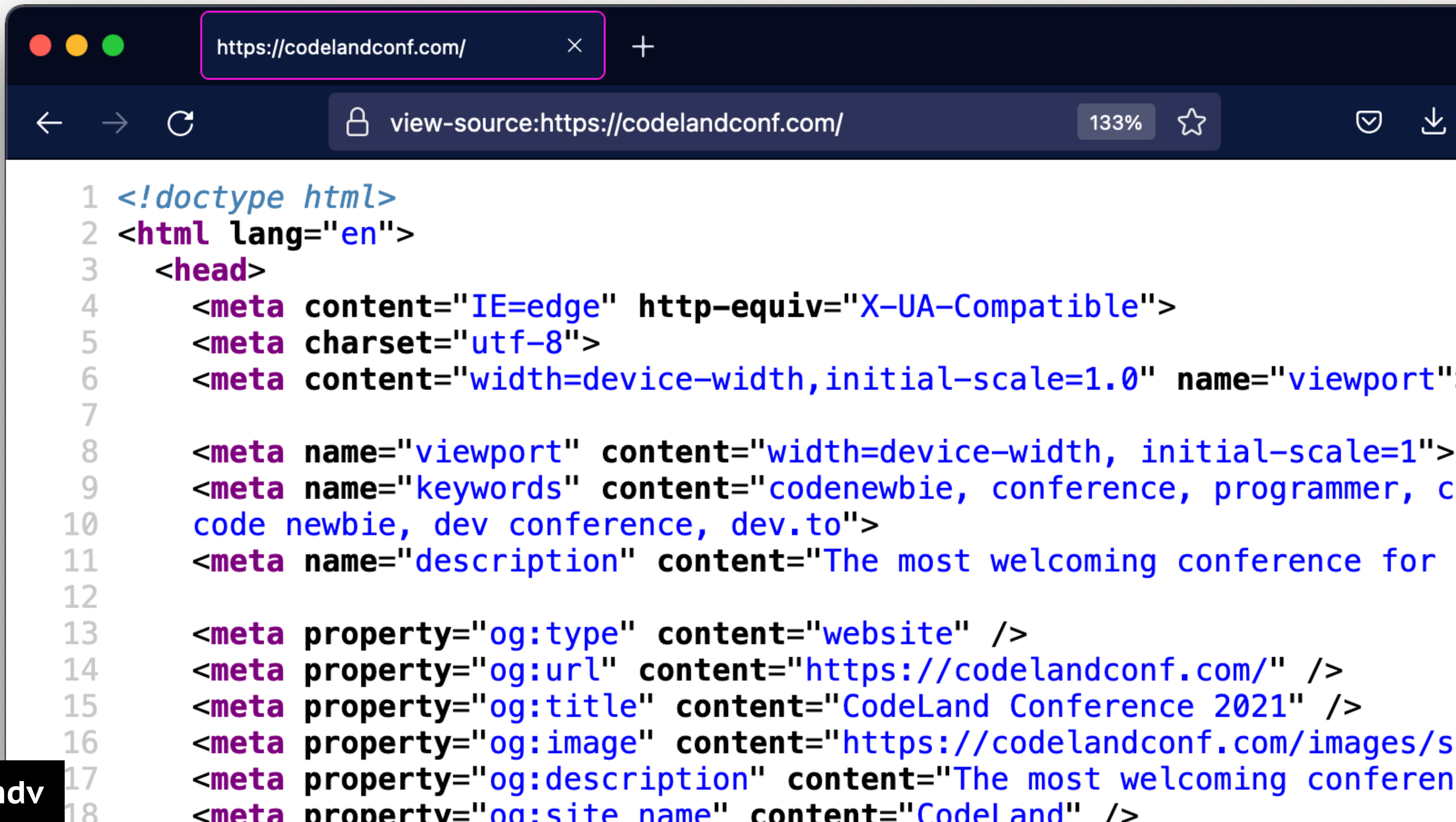
https://codelandconf.com

133%



CodeLand





```
1 <!doctype html>
2 <html lang="en">
3   <head>
4     <meta content="IE=edge" http-equiv="X-UA-Compatible">
5     <meta charset="utf-8">
6     <meta content="width=device-width,initial-scale=1.0" name="viewport">
7
8     <meta name="viewport" content="width=device-width, initial-scale=1">
9     <meta name="keywords" content="codenewbie, conference, programmer, c
10    code newbie, dev conference, dev.to">
11    <meta name="description" content="The most welcoming conference for
12
13    <meta property="og:type" content="website" />
14    <meta property="og:url" content="https://codelandconf.com/" />
15    <meta property="og:title" content="CodeLand Conference 2021" />
16    <meta property="og:image" content="https://codelandconf.com/images/s
17    <meta property="og:description" content="The most welcoming conferen
18    <meta property="og:site name" content="Codeland" />
```

CodeLand

Menu

CodeLand

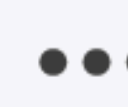
2021

@hdv

most welcoming conference for



Inspector



Search HTML



<!DOCTYPE html>

<html lang="en"> event scroll overflow

▶ <head> ... </head>

▼ <body class="index">

▼ <nav class="site-nav"> flex

▶ <div class="nav-link--container"> ...

</div> flex

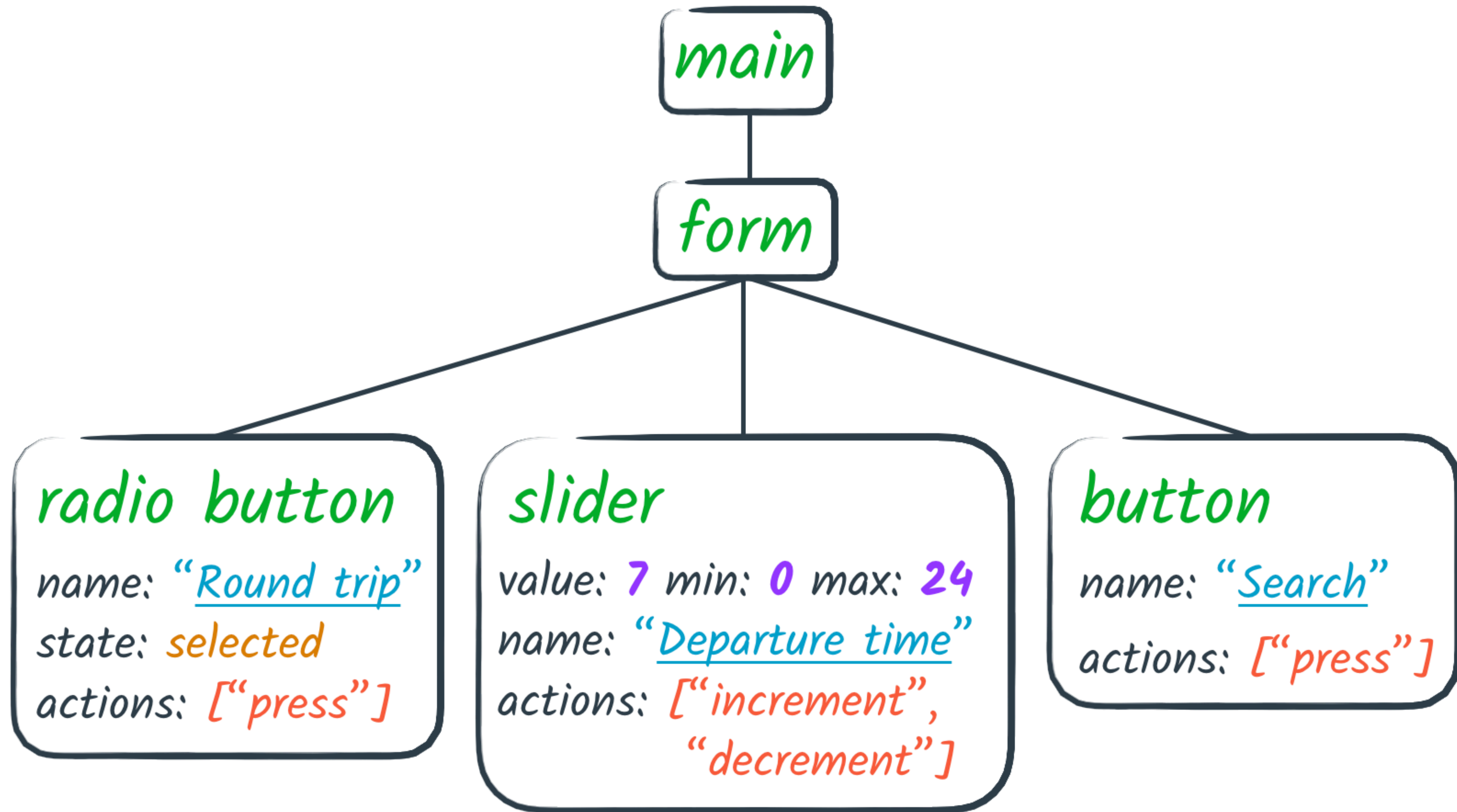
</nav>

▼ <main>

▶ <section class="hero-section" aria-label="Introduction to CodeLand 2021"> ... </section> flex

▶ <section id="overview" class="section-card overview" aria-label="CodeLand 2021 Overview"> ... </section>

▶ <section id="attend" class="section-card attend" aria-labelledby="attendHeader"> ... </section>



Role

What kind of thing is it?

Name/Description

How should we refer to it?

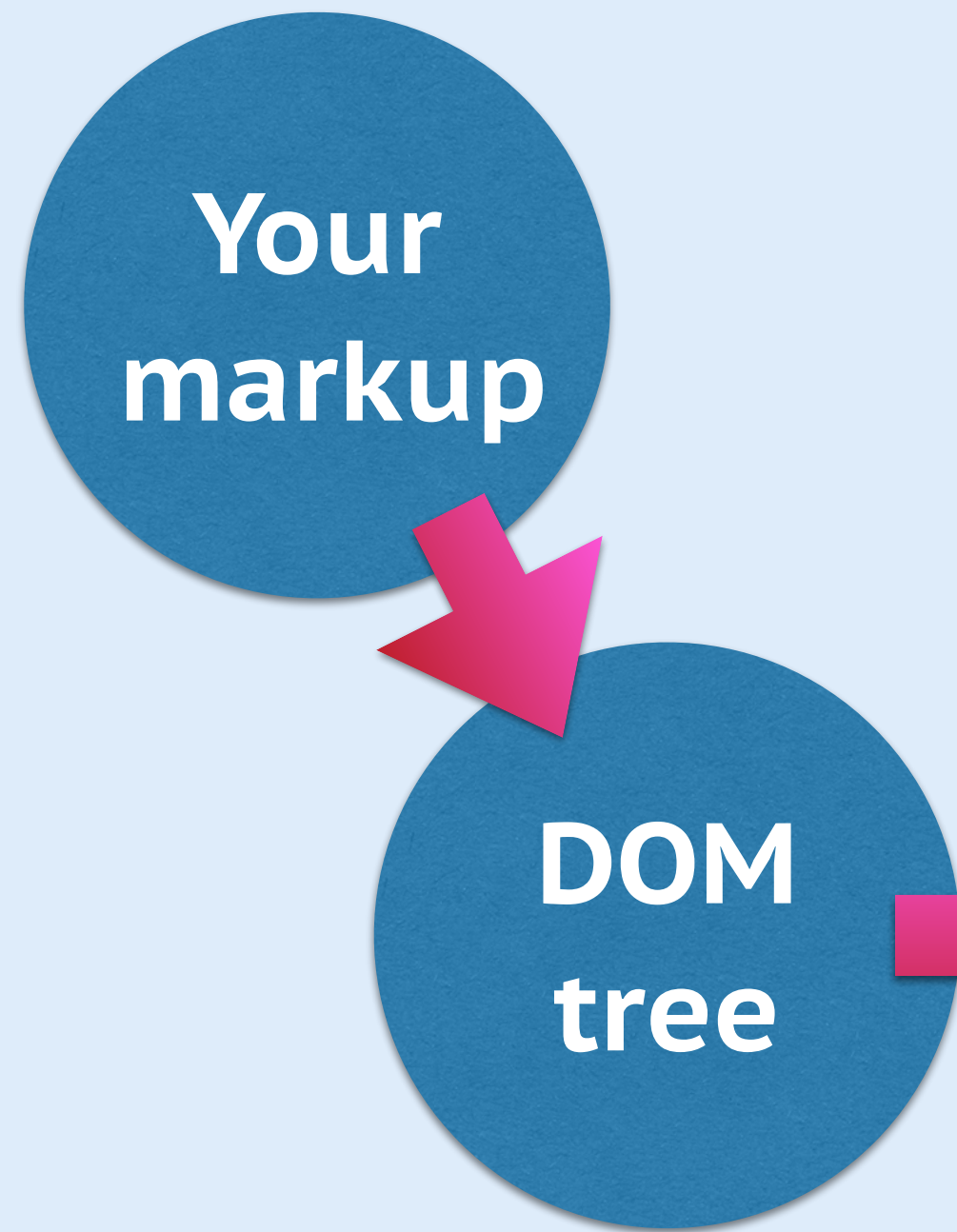
State/properties

What else should we know?

**Accessible code points
machines at what stuff is**

**Accessible code points
machines at what stuff is called**

**Accessible code points
machines in what state stuff is**



Accessibility tree

A screenshot of the Accessibility Inspector tool. The top bar shows 'Inspector' and 'Accessibility' tabs. Below the bar, there's a toggle for 'Turn Off Accessibility Features'. The main area displays the accessibility tree with columns for 'Role' and 'Name'. The tree structure is as follows:

- Role: (empty), Name: "(@mozilla)"
- statictext: ""
- list: ""
 - listitem: "Twitter (@mozilla)"
 - link: "Twitter (@mozilla)" (highlighted)
 - text leaf: ""
 - listitem: "Instagram (@mozilla)"
 - text leaf: ""
- section: ""
 - heading: "Firefox"

Below the tree, the 'Properties' section for the selected 'link' role is shown:

- name: "Twitter (@mozilla)"
- role: "link"
- actions: [...]
- value: "https://twitter.com/mozilla"
- DOMNode: a.twitter
- description: ""
- keyboardShortcut: ""
- childCount: 2
- indexInParent: 0
- states: [...]
- attributes: {...}

Platform APIs

Microsoft

Microsoft Active Accessibility
Microsoft User Interface
Automation
MSAA

Apple

Mac OS X Accessibility
Protocol

THE LINUX FOUNDATION GNOME DEVELOPER

Linux/Unix Accessibility
Toolkit
IAccessible2

AT

JAWSforWindows

VoiceOver

NVDA

text-to-speech
screen magnifiers
alternate pointing devices



**It's the markup
that matters**

What's this page?

```
<html lang="nl">  
  <head>  
    <title>It's the markup that matters –  
    Presentations – My site</title>  
  </head>  
  <body>  
    ...  
  </body>  
</html>
```

“

[The <title>] is still the first guarantee or first confirmation that you've ended up on the page that you intended to reach.

– Léonie Watson, accessibility expert and screenreader user

What's on this page?

Headings

“Navigate by heading” is a very common way for AT users to figure out what's on a page



Assistive technology

From Wikipedia, the free encyclopedia
(Redirected from [Assistive technologies](#))

Assistive technology is an [umbrella term](#) that includes assistive, adaptive, and rehabilitative devices for [people with disabilities](#) while also including the process used in selecting, locating, and using them. People who have disabilities often have difficulty performing [activities of daily living](#) (ADLs) independently, or even with assistance. ADLs are self-care activities that include toileting, mobility (ambulation), eating, bathing, dressing and grooming.



aring aid

"reduce inst

Headings

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](#)

What's on this page?

Landmarks

Landmarks let you specify which sections a page has, so that people can navigate to them

HTML5

<header>

<footer>

<nav>

<aside>

<main>

ARIA roles

role="banner"

role="contentinfo"

role="navigation"

role="complementary"

role="main"

“

Built-in beats bolt-on.

– Bruce Lawson

HTML5

<header>

<footer>

<nav>

<aside>

<main>

Form controls

“

“59% of form inputs were not properly labeled.”.

— WebAIM

**Label input fields, radio buttons,
checkboxes, selects, textareas, etc...
so that they have a *name***

First Name

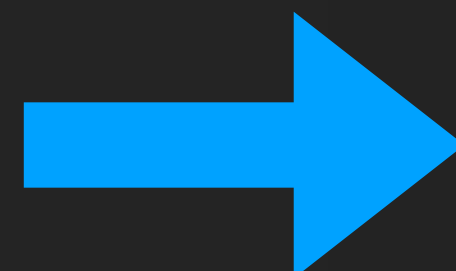
```
<div class="form-item">  
  First Name  
  <input type="text" />  
</div>
```

Role: 'textbox' / 'entry'
Name: null / ''

Name: ""

aria-labelledby: *Not specified*

aria-label: *Not specified*

 **From label:** *Not specified*

placeholder: *Not specified*

aria-placeholder: *Not specified*

title: *Not specified*

First Name

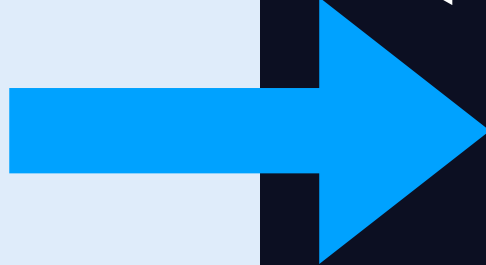
```
<div class="form-item">  
  <label for="fn">First Name</label>  
  <input id="fn" type="text" />  
</div>
```

Role: 'textbox' / 'entry'
Name: 'First Name'

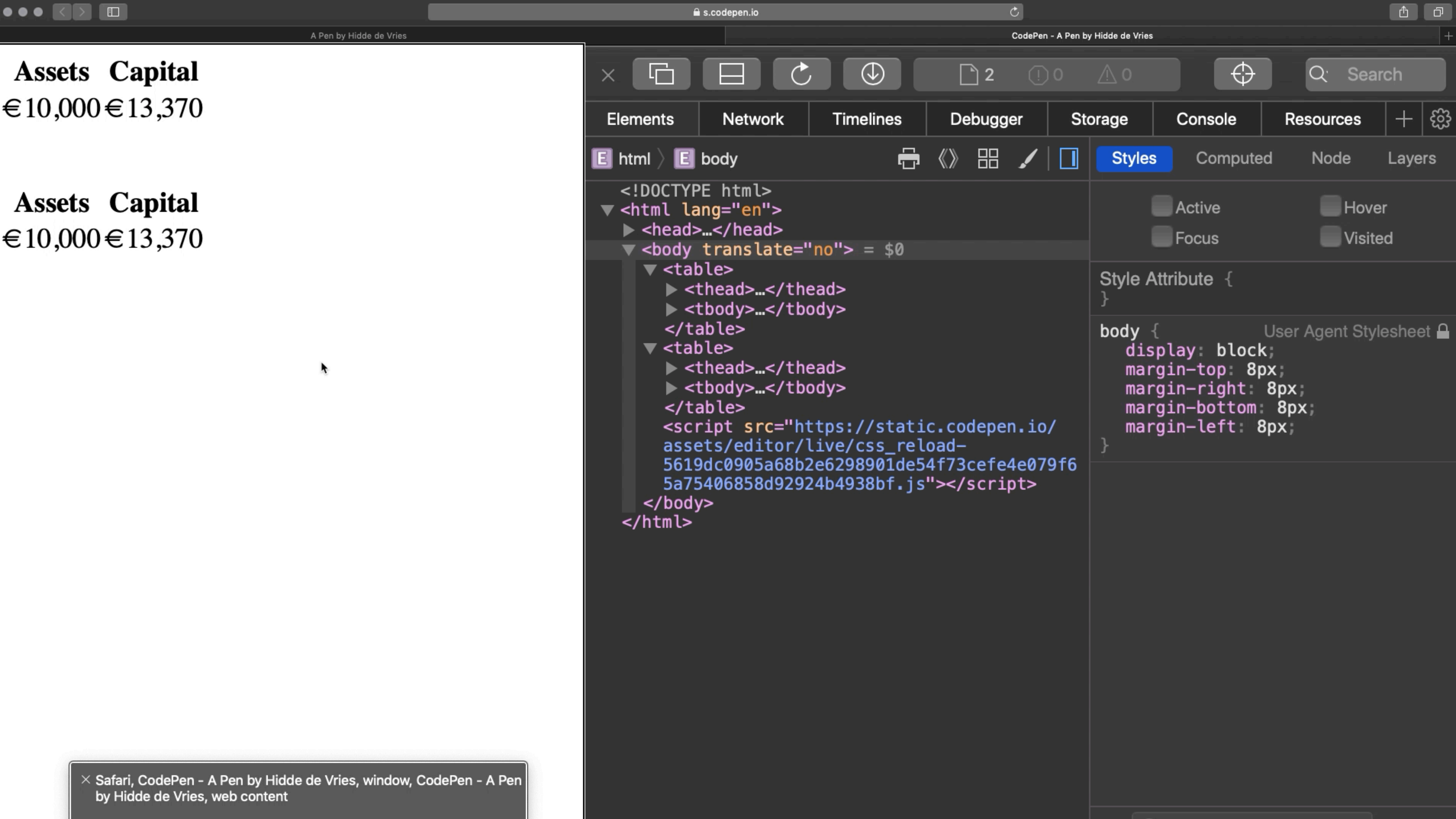
Table semantics

**Tables make tabular data
easier to navigate**

Use table semantics



```
<table>
  <caption>Net results 2018</caption>
  <thead>
    <tr>
      <th scope="col">Assets</th>
      <th scope="col">Capital</th>
      ...
    </tr>
  </thead>
  <tbody>
    ...
  </tbody>
</table>
```

Assets	Capital
€10,000	€13,370

Assets	Capital
€10,000	€13,370

```
Elements Network Timelines Debugger Storage Console Resources +
html > body
<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body translate="no"> = $0
    <table>
      <thead>...</thead>
      <tbody>...</tbody>
    </table>
    <table>
      <thead>...</thead>
      <tbody>...</tbody>
    </table>
    <script src="https://static.codepen.io/assets/editor/live/css_reload-5619dc0905a68b2e6298901de54f73cefe4e079f65a75406858d92924b4938bf.js"></script>
  </body>
</html>

Style Attribute {
}

body {
  display: block;
  margin-top: 8px;
  margin-right: 8px;
  margin-bottom: 8px;
  margin-left: 8px;
}
```

Use table semantics



```
<table>  
  <caption>Net results 2018</caption>  
  <thead>  
    <tr>  
      <th scope="col">Assets</th>  
      <th scope="col">Capital</th>  
      ...  
    </tr>  
  </thead>  
  <tbody>  
    ...  
  </tbody>  
</table>
```

Language

**The lang attribute lets you specify
which language content is in**

Specify languages with lang

```
<div>  
  <p lang="de">Willkommen</p>  
  <p lang="fr">Bienvenue</p>  
  <p lang="en">Welcome</p>  
</div>
```

If buttons are just icons



The “Add Reaction” button



```
<div class="icon" onclick="emoji()">
  <svg class="icon">
    <use xlink:href="#emoji"></use>
  </svg>
</div>
```

Role: "
Name: "



```
// use the button element
<button type="button">
  <svg class="icon">
    <use xlink:href="#emoji"></use>
  </svg>
</button>
```

Role: 'button'
Name: ''



```
<button type="button">
  <svg class="icon">
    <use xlink:href="#emoji"></use>
  </svg>
  // add accessible name
  <span class="visually-hidden">
    Add reaction
  </span>
</button>
```

Role: 'button'
Name: 'Add reaction'



```
<button type="button">
  // hide icon from a11y tree
  <svg
    class="icon"
    role="presentation"
    aria-hidden="true"
    focusable="false">
    <use xlink:href="#emoji"></use>
  </svg>
  <span class="visually-hidden">
```

*“Accessibility node
not exposed”*

“

It would appear that either `aria-hidden="true"` or `role="presentation"` should be able to help out here.

— John Foliot



```
<button type="button">
  <svg
    class="icon"
    role="presentation"
    aria-hidden="true"
    focusable="false">
    <use xlink:href="#emoji"></use>
  </svg>
  <span class="visually-hidden">
    Add reaction
```

*Avoids weird focus
bugs in IE11*

**No meaningful mark-up for your
component? ARIA can polyfill!**

**ARIA can provide names,
roles, property and states
through attributes.**

**There usually *is* existing HTML
for your problem, combining
existing elements FTW!**

Not all ARIA has good support or provides good usability for AT users.

Expandables

Tooltip

Enter your IBAN number ?

An IBAN-number is a 34 characters string, please double check with your bank if you're not sure what yours is.

```
<button  
  type="button"  
  aria-controls="tooltip"  
>
```

What's IBAN?

```
</button>
```

```
<div id="tooltip">
```

An IBAN-number is a 34 ch...

```
</div>
```


Tooltip

Enter your IBAN number ?

An IBAN-number is a 34 characters string, please double check with your bank if you're not sure what yours is.

```
<button  
  type="button"  
  aria-controls="tooltip"  
  aria-expanded="true"  
>
```

What's IBAN?

```
</button>
```

```
<div id="tooltip">
```

An IBAN-number is a 34 ch...

```
</div>
```

Tooltip

Enter your IBAN number ?

```
<button  
  type="button"  
  aria-controls="tooltip"  
  aria-expanded="false"  
>
```

What's IBAN?

```
</button>
```

```
<div id="tooltip" hidden>
```

An IBAN-number is a 34 ch...

```
</div>
```

Announcements



Your changes were saved successfully!

```
<div role="alert">  
</div>
```

```
// We've just used fetch() to POST form  
// data on the submit event of a form.  
// Let's say we're handling like:
```

```
const el = document.querySelector("div");
```

```
el.textContent("Your changes were saved  
successfully");
```



Your changes were saved successfully!

Meaning without mark-up: the Accessibility Object Model



“This effort aims to develop additions to the web platform to allow developers to provide information to assistive technology APIs, and to understand what information browsers provide to those APIs.”

– The AOM explainer document

AOM doesn't bring new semantics or states, it introduces new ways to specify them

No more
“sprouting”

```
<my-custom-element  
  role="button"  
  aria-disabled="false"  
  
  ...  
>
```


Set ARIA with IDL attributes instead of markup

```
const e1 = document.querySelector("e1");  
  
e1.role = "button";  
e1.ariaDisabled = false;
```

Relationships without IDREFs

Events from Assistive Technologies

```
// Implementing a canvas-based
// spreadsheet's semantics
canvas.attachAccessibleRoot();

let table = canvas.accessibleRoot
    .appendChild(new AccessibleNode());

table.role = "table";
table.colCount = 10;
table.rowcount = 100;

let headerRow = table.appendChild(
    appendChild(new AccessibleNode())
);
headerRow.role = "row";
```

Non-DOM nodes in the Accessibility Tree

Reading accessibility tree through JavaScript

TLDR

Our apps need to be accessible

Good mark-up makes a difference, it gives us lots of accessibility for free

Always consider which names, roles and states your markup conveys