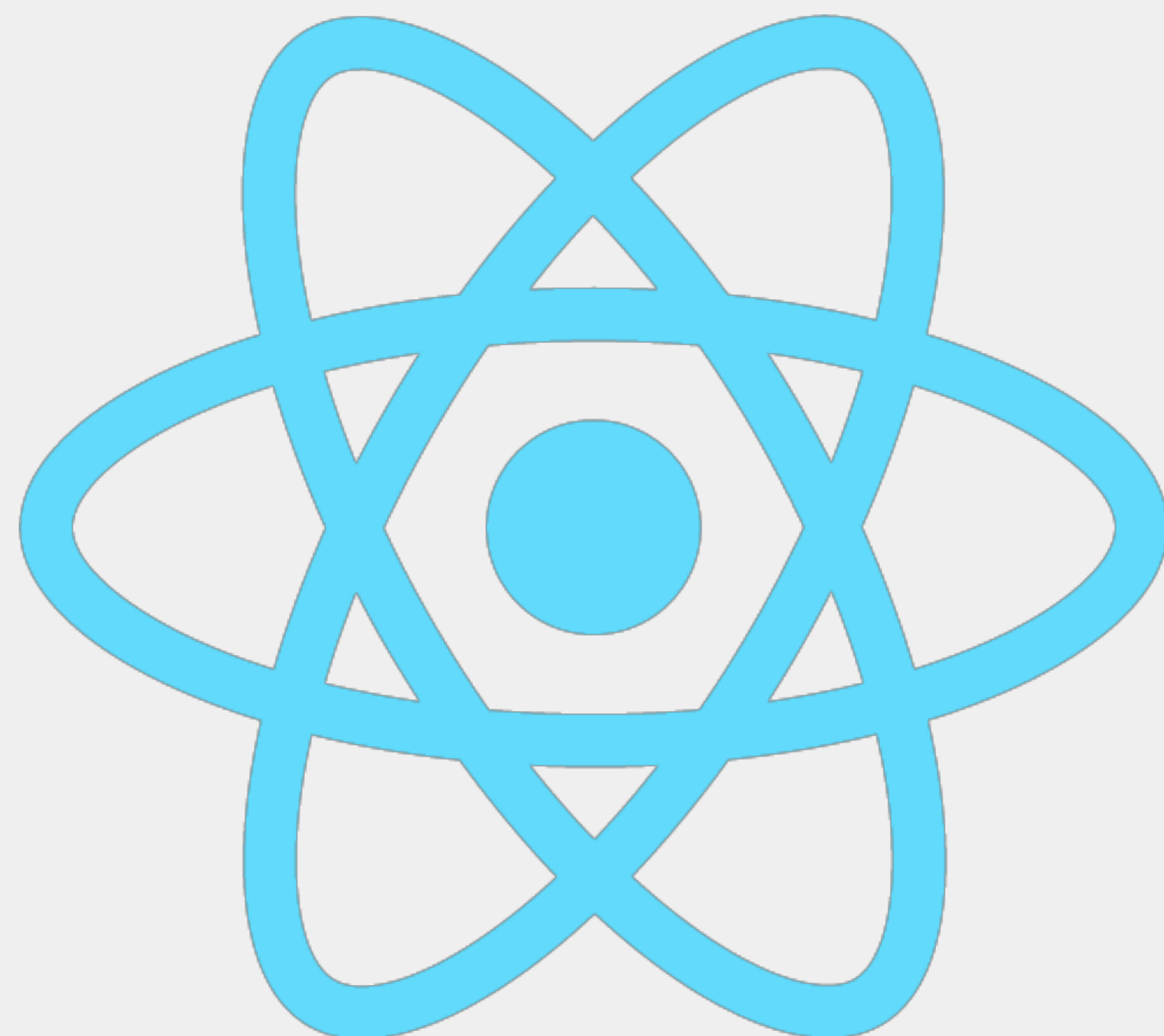






Manuel Matuzović

Frontend Developer from Vienna

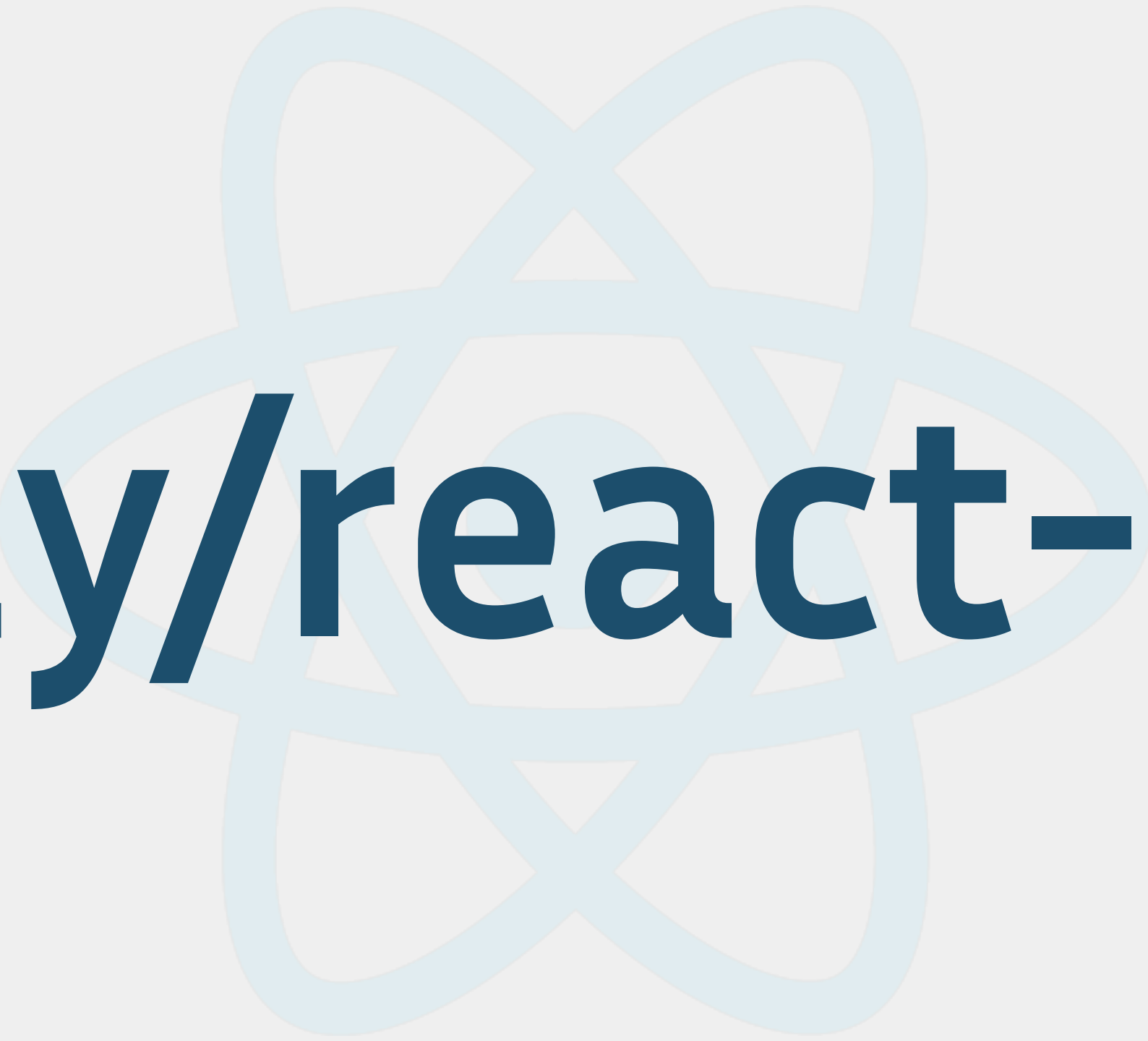


City of
Vienna

HTML, CSS, Accessibility

twitter: @mmatuzo





bit.ly/react-tips



12 Tips For More Accessible React Apps





⁸ ~~12~~ Tips For More Accessible React Apps



a11y tip #1

Create a sound
document outline.

a11y tip #1

<h1>Yo! I'm the title of your page.</h1>

<h2>I'm very important.</h2>

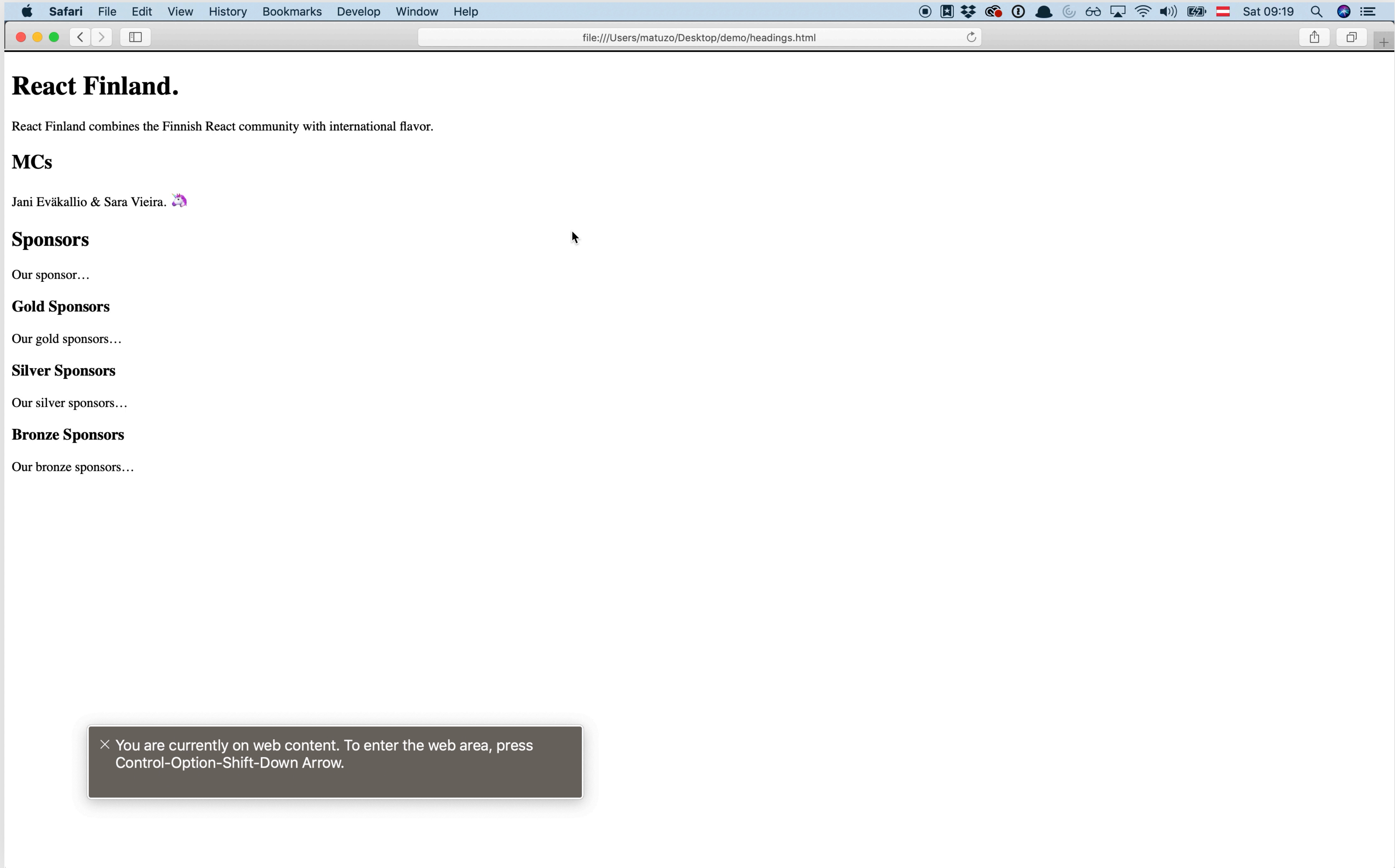
<h3>My parent is very important.</h3>

<h3>My parent is very important.</h3>

<h4>I exist.</h4>

<h2>I'm very important.</h2>

<h3>My parent is very important.</h3>



a11y tip #1

```
<Heading.H>I will be an h1</Heading.H>
```

```
<Heading.LevelBoundary>
```

```
  <Heading.H>I will be an h2</Heading.H>
```

```
  <Heading.LevelBoundary>
```

```
    <Heading.H>I will be an h3</Heading.H>
```

```
  </Heading.LevelBoundary>
```

```
    <Heading.H>I will be an h2</Heading.H>
```

```
  </Heading.LevelBoundary>
```

a11y tip #1

<h1>I will be an h1</h1>

<h2>I will be an h2</h2>

<h3>I will be an h3</h3>

<h2>I will be an h2</h2>



Eduskuntavaalien virallinen vaalipäivä on sunnuntaina 14. huhtikuuta. Vaalipäivänä kaikki äänestyspaikat ovat avoinna kello 9–20. Vaalipäivänä saa äänestää vain äänioikeusilmoitukseen merkityssä äänestyspaikassa.

Transferring data from eu1.snoobi.com...

12.04.2019 10:00

a11y tip #1

Create a sound document outline.

- It gives your document structure.
- Helps screen reader users with navigation.
- Important for SEO.
- Check out **Tenon UI's headings** component.
- Test your document outline with **tota11y** or **wave**.

```
<h1>React Finland.</h1>
```

```
<h2>MCs</h2>
```

```
<h2>Speakers</h2>
```

```
<h2>Sponsors</h2>
```

```
<h3>Gold Sponsors</h3>
```

```
<h3>Silver Sponsors</h3>
```

```
<h3>Bronze Sponsors</h3>
```


a11y tip #2

Hide content
correctly

orkshop

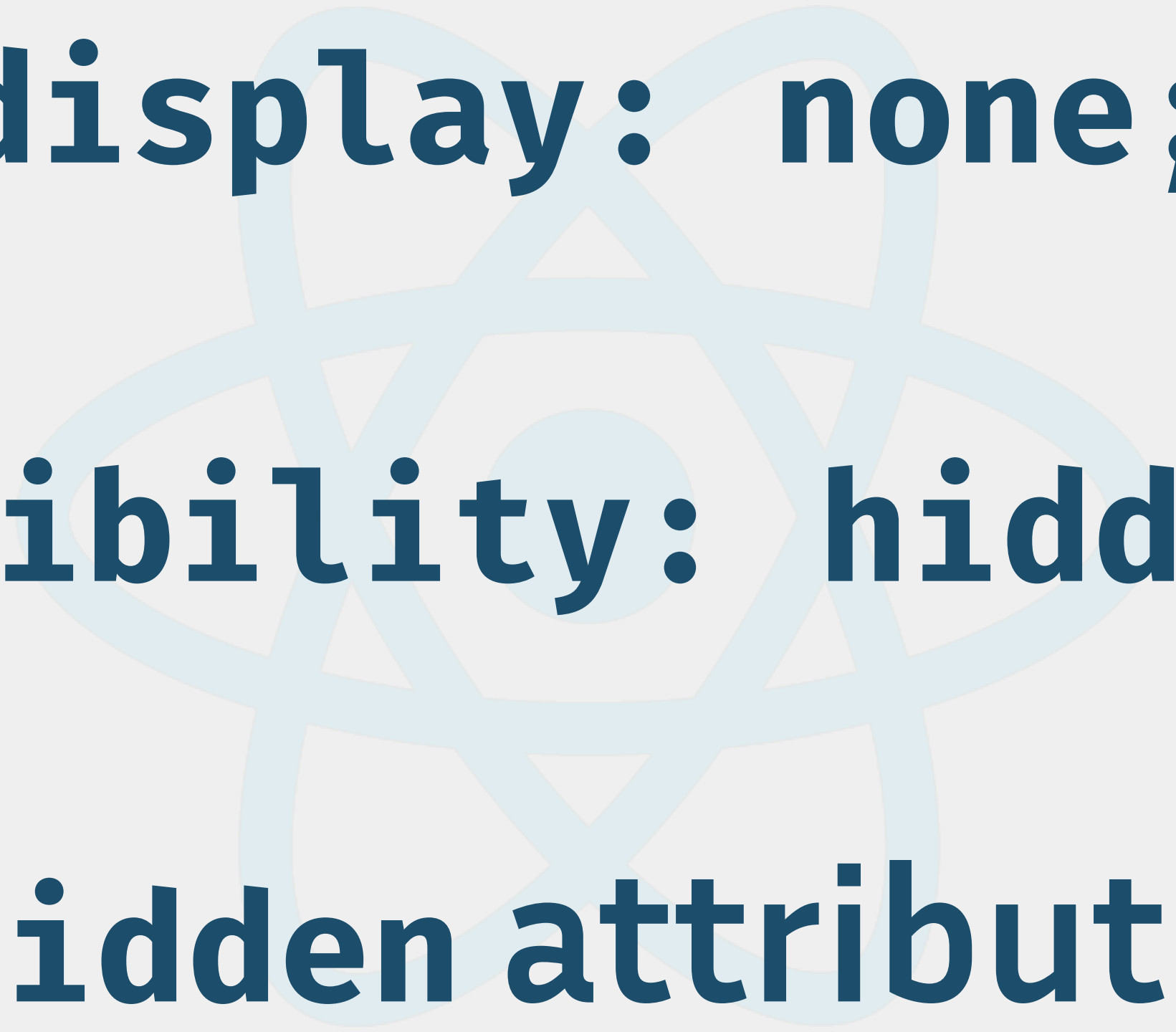
Headings

Annotate:  

- 1 React Finland
 - 2 Learn More about React, Explore Finland.
 - 3 Till the event:
 - 2 MCs#
 - 2 Workshop Instructors#
 - 2 Speakers#
 - 2 Tickets#
 - 2 🏆 Gold Sponsors#
 - 2 🥈 Silver Sponsors#
 - 2 🥉 Bronze Sponsors#

Summary

nsors



display: none;
visibility: hidden;
hidden attribute



~~display: none;~~

~~visibility: hidden;~~

~~hidden attribute~~



a11y tip #2

```
.visually-hidden {  
  border: 0;  
  clip: rect(0 0 0 0);  
  height: 1px;  
  margin: -1px;  
  overflow: hidden;  
  padding: 0;  
  position: absolute;  
  width: 1px;  
  white-space: nowrap;  
}
```



React Finland.

MCs

Speakers

Gold Sponsors

Silver Sponsors

Bronze Sponsors

```
<h1>React Finland.</h1>
```

```
<h2>MCs</h2>
```

```
<h2>Speakers</h2>
```

```
<h2 className="visually-hidden">  
  Sponsors  
</h2>
```

```
<h3>Gold Sponsors</h3>
```

```
<h3>Silver Sponsors</h3>
```

```
<h3>Bronze Sponsors</h3>
```



a11y tip #2

```
<button>  
  <span class="visually-hidden">Save</span>  
  
  <svg aria-hidden="true" width="32" height="32">  
    <path ...></path>  
  </svg>  
</button>
```


a11y tip #2

```
<button>  
  <VisuallyHidden>Save</VisuallyHidden>  
  
  <svg aria-hidden="true" width="32" height="32">  
    <path ...></path>  
  </svg>  
</button>
```

<https://github.com/reach/reach-ui/tree/master/packages/visually-hidden>

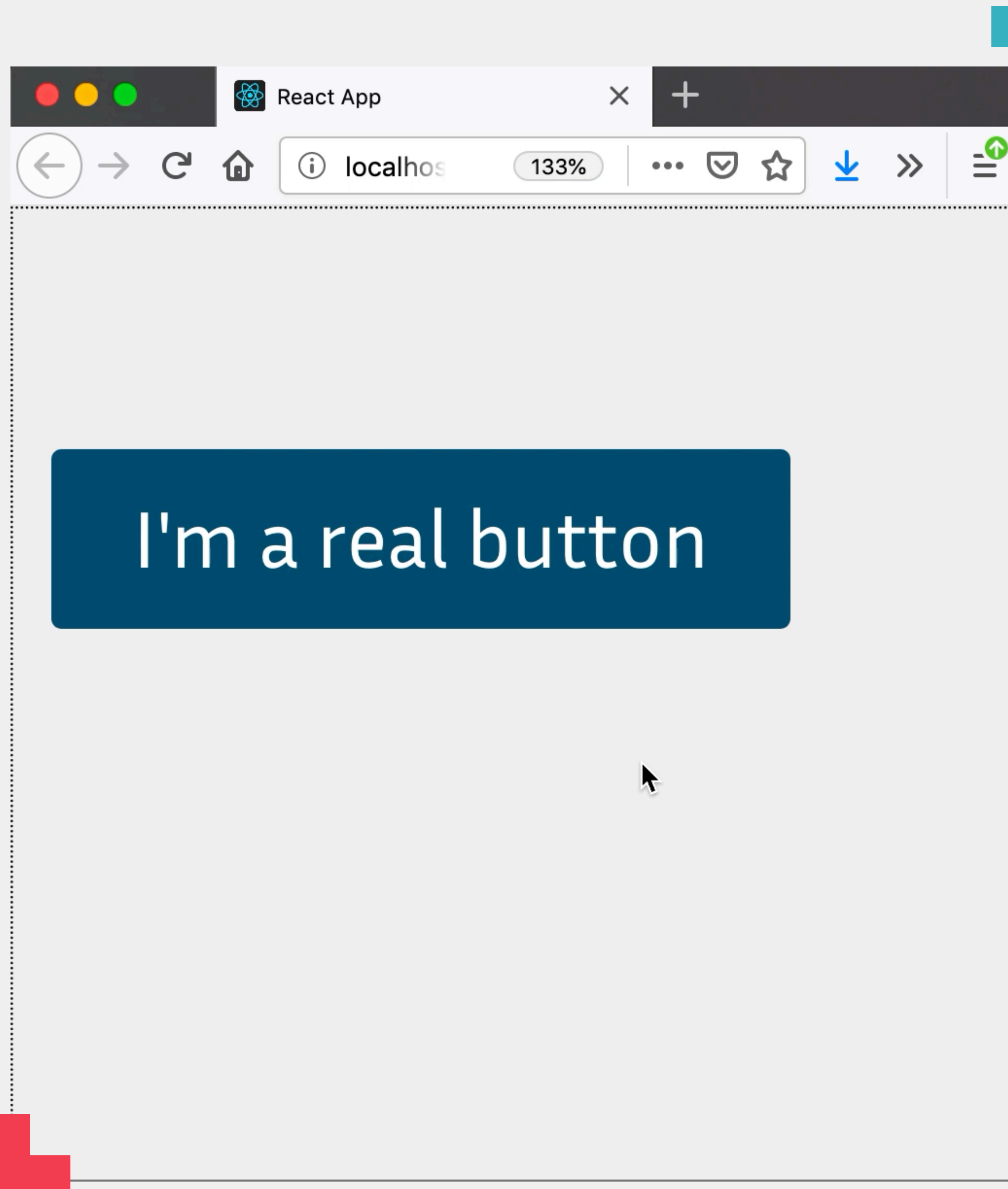
Hide content correctly

- **display: none; visibility: hidden;** and the **hidden** attribute remove content from the accessibility tree.
- Every item needs a textual representation, even if it isn't visible.
- Check out **Reach UI's VisuallyHidden** component.

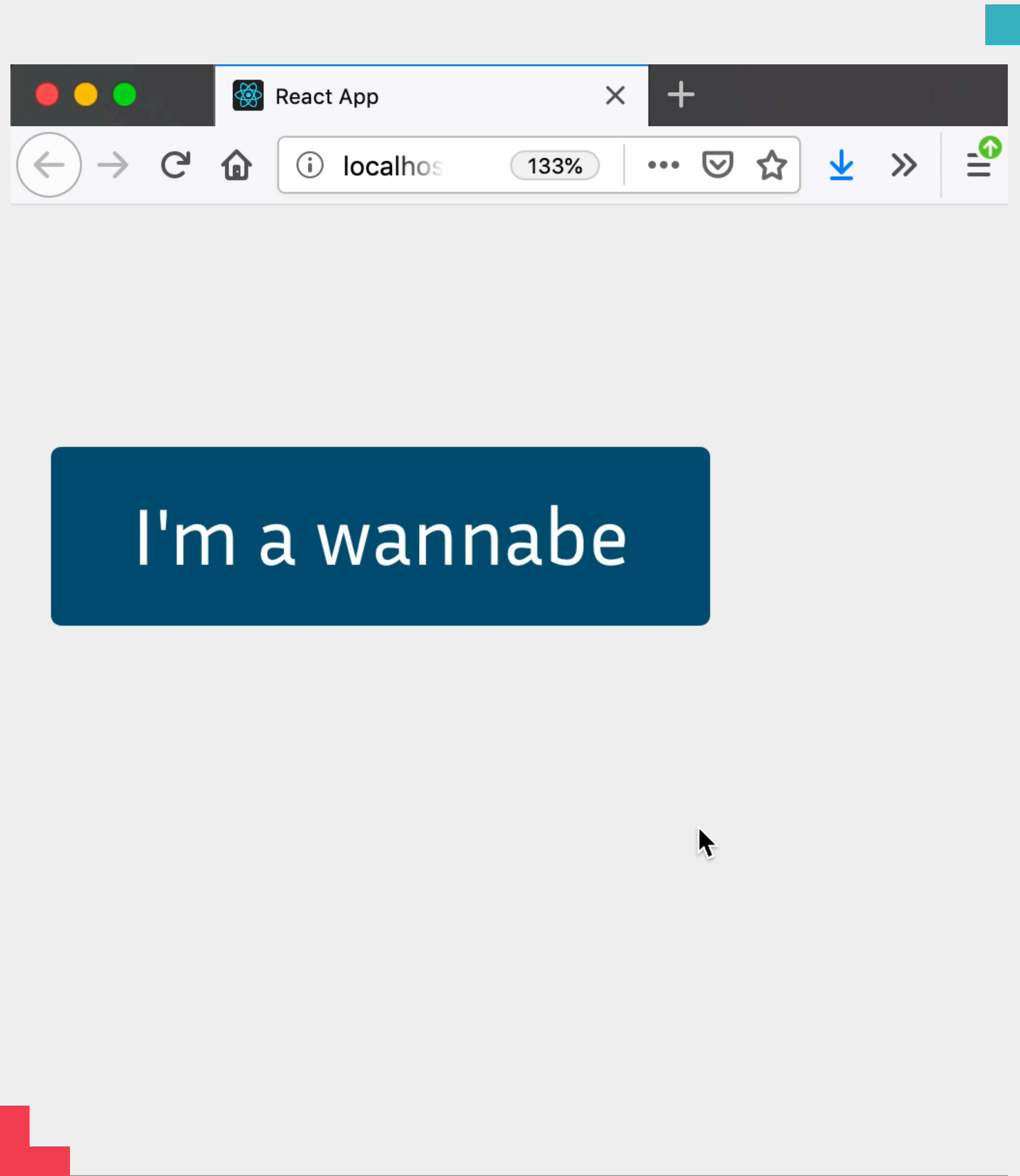
```
.visually-hidden {  
  border: 0;  
  clip: rect(0 0 0 0);  
  height: 1px;  
  margin: -1px;  
  overflow: hidden;  
  padding: 0;  
  position: absolute;  
  width: 1px;  
  white-space: nowrap;  
}
```

a11y tip #3

Use **<button>** if
you need a button



```
<button  
  className="btn"  
  onClick={this.handleClick}>  
    I'm a real button  
</button>
```



```
<div  
  className="btn"  
  onClick={this.handleClick}>  
    I'm a wannabe  
</div>
```

Use `<button>` if you need a button

- `<button>`s are focusable by default.
- They come with keyevents for free.
- `<button>`s are semantic. A `<div>` is just generic text.
- Check out [Just use button](#) by Rob Dodson and [The Links vs. Buttons Showdown](#) by Marcy Sutton on YouTube.

`<button>`



`</button>`

a11y tip #4

Use fragments to
avoid invalid HTML

a11y tip #4

```
const Table = props => {  
  return (  
    <table>  
      <tr>  
        <Columns />  
      </tr>  
    </table>  
  );  
}
```


a11y tip #4

```
const Columns = props => {  
  return (  
    <div>  
      <td>Hello</td>  
      <td>World</td>  
    </div>  
  );  
}
```

a11y tip #4



```
<table>
  <tr>
    <div>
      <td>Hello</td>
      <td>World</td>
    </div>
  </tr>
</table>
```



a11y tip #4

```
const Columns = props => {  
  return (  
    <React.Fragment>  
      <td>Hello</td>  
      <td>World</td>  
    </React.Fragment>  
  );  
}
```


a11y tip #4



```
<table>  
  <tr>  
    <td>Hello</td>  
    <td>World</td>  
  </tr>  
</table>
```



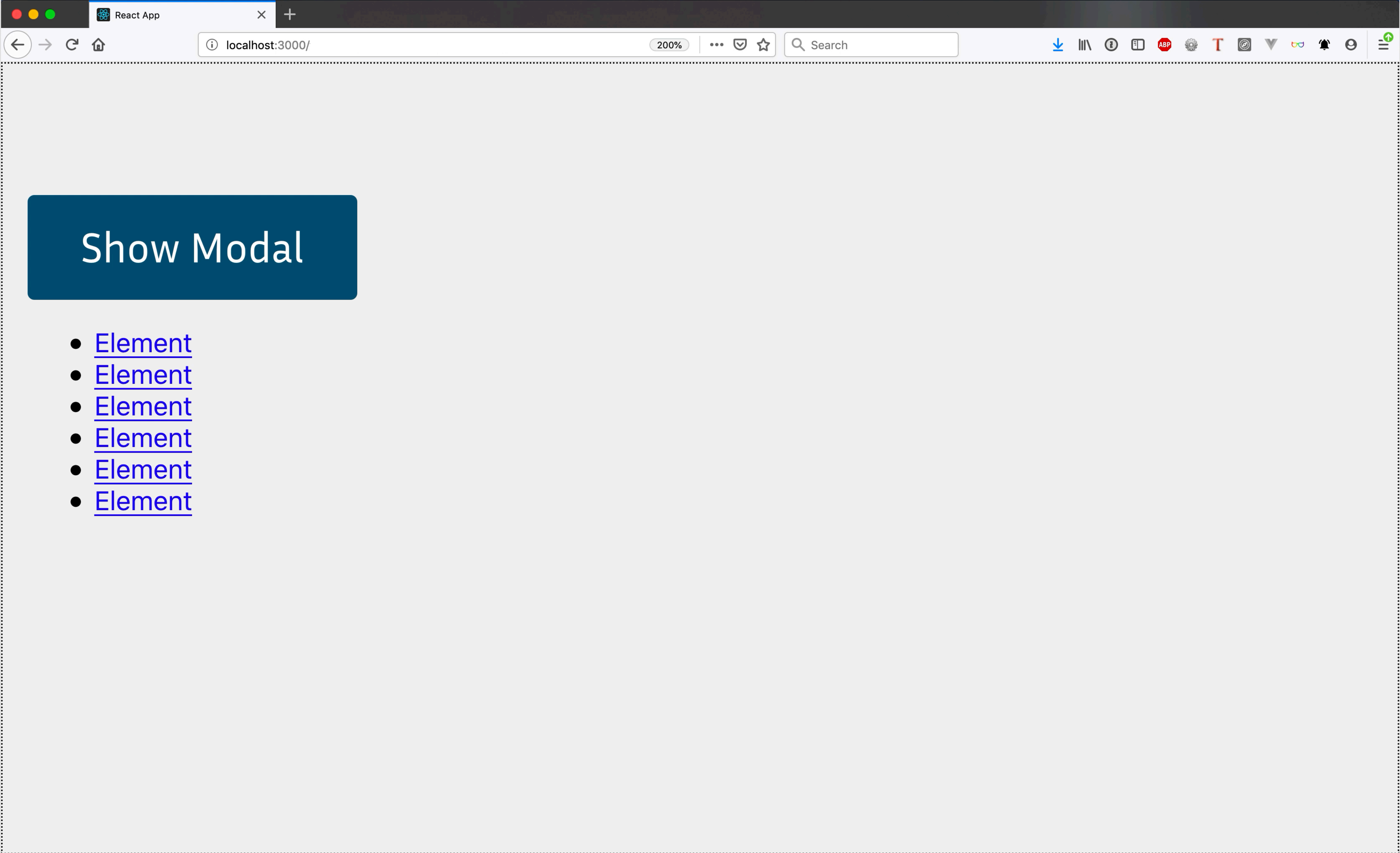
Use fragments to avoid invalid HTML

- Fragments help you write valid HTML.
- They reduce bloat.
- There's also a shorter syntax (see code example ➡)
- Check out the **Fragments docs** for more details and examples.

```
function Columns() {  
  return (  
      
    <td>Hello</td>  
    <td>World</td>  
    </>  
  );  
}
```

a11y tip #5

Take care of focus
management.





- Element
- Element
- Element
- Element
- Element
- Element

a11y tip #5

```
class Button2 extends React.Component {  
  constructor(props) {  
    super(props);  
    this.btn = React.createRef();  
  }  
  
  setFocus(){  
    this.btn.current.focus();  
  }  
  
  render() {  
    return (  
      <button className="btn" ref={ this.btn }>  
        { this.props.children }  
      </button>  
    )  
  }  
}
```



Firefox NightlyFileEditViewHistoryBookmarksToolsWindowHelp

VICE - Unbequemer JournalismX

https://www.vice.com/de_at

Search

8

☰

VICE

VICE-Kanäle

CANNABIS

Wir haben herausgefunden, welcher Rapper am meisten über Drogen spittet

Ob Kollegah, 187 Strassenbande, RAF Camora oder Samy: Deutsche Rapper lieben Drogen. Zumindest in ihren Songs. Doch wer hat die meisten Lines über Lines?

KILIAN BÜHLING & MARCUS STAIGER

Neu

BIENEN IM AUGE

RASSISMUS

DROGEN

DROGEN

Waiting for video-images.vice.com...



Take care of focus management.

- Focus management is important because it's essential for keyboard and screen reader users.
- Take advantage of `refs` in React to manage focus.
- Check out the [A11y dialog](#) on Github and the [accessibility docs](#) on [reactjs.org](#).

```
class CustomTextInput extends React.Component {  
  constructor(props) {  
    super(props);  
    // Create a ref to store the textInput  
    // DOM element  
    this.textInput = React.createRef();  
  }  
  render() {  
    // Use the `ref` callback to store a  
    // reference to the text input DOM  
    // element in an instance field  
    // (for example, this.textInput).  
    return (  
      <input  
        type="text"  
        ref={this.textInput}  
      />  
    );  
  }  
}
```


a11y tip #6

Make notifications
accessible to everyone



Safari

File

Edit

View

History

Bookmarks

Develop

Window

Help



localhost



Save

× React App, web content

a11y tip #5

```
<div className="alert" role="alert">  
  Saved successfully  
</div>
```




Safari

File

Edit

View

History

Bookmarks

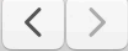
Develop

Window

Help



Mon 09:05



localhost



Save

× React App, web content

Make notifications accessible to everyone

- Use `role="alert"` or `role="status"` to create live regions.
- Use live regions only for important information.
- Check out **Reach UI's alert** component.
- Check out **react-aria-live** on Github.
- Read **ARIA live regions in React** on Almero Steyns blog.

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

```
  Saved successfully
```

```
</div>
```

```
// alert will interrupt the screen reader if  
it's in the course of announcing something  
else.
```

```
<div className="alert" role="status">
```

```
  Saved successfully
```

```
</div>
```

```
// status will wait until the screen reader has  
finished announcing.
```

```
<div className="alert" role="alert" aria-  
atomic="true">
```

```
  Saved successfully
```

```
  <span>I will be announced as well</span>
```

```
</div>
```

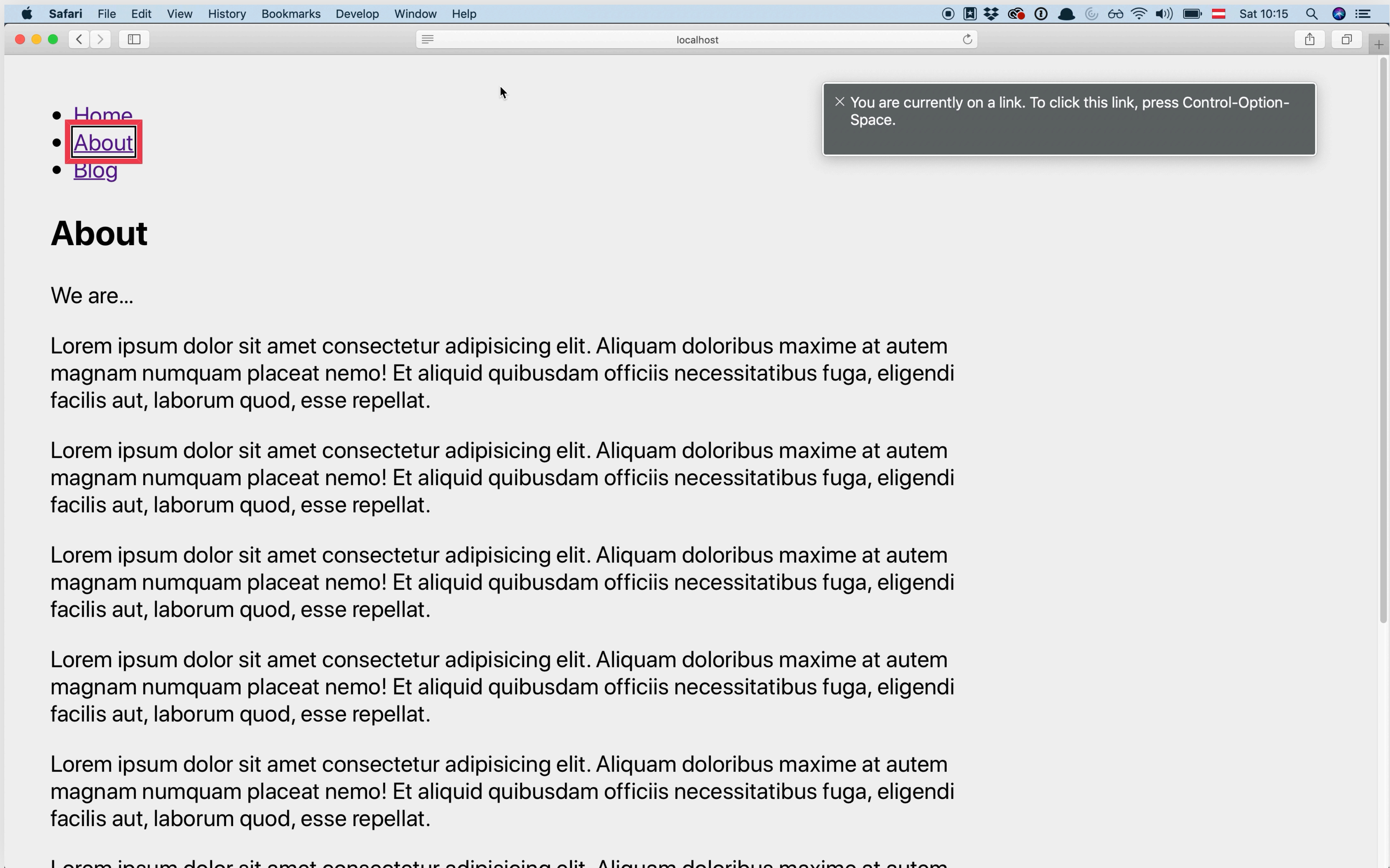
```
// aria-atomic="true" - announces everything not  
just the content that has changed.
```


a11y tip #7

Announce page changes



A screenshot of a web browser window. The browser's address bar shows 'localhost'. The page has a light gray background. In the top left, there is a navigation menu with three items: 'Home' (highlighted with a red box), 'About', and 'Blog'. Below this menu is a large heading 'Home'. Under the heading is a paragraph of Lorem Ipsum text. At the bottom of the page, there is a section titled 'Footer Links' with three links: 'Home', 'Blog', and 'About'. In the top right corner, there is a dark gray tooltip box with the text '× visited, link, Home list 3 items'. The browser's top bar shows various icons and the time 'Sat 10:15'.



a11y tip #7

```
class About extends React.Component {
  constructor(props) {
    super(props);
    this.section = React.createRef();
  }
  componentDidMount(){
    this.section.current.focus();
    document.title = "About";
  }

  render() {
    return (
      <section tabIndex="-1" ref={ this.section }>
        <h2>About</h2><p>We are...</p><p>Lorem ipsum...</p>
      </section>
    )
  }
}
```

a11y tip #7

```
render() {  
  return (  
    <section aria-labelledby="pageTitle" tabIndex="-1" ref= { this.section }>  
      <h2 id="pageTitle">About</h2>  
      <p>We are...</p><p>Lorem ipsum...</p>  
    </section>  
  )  
}
```

SafariFileEditViewHistoryBookmarksDevelopWindowHelplocalhost

Home

About

Blog

Home

The homepage...

Lorem ipsum dolor sit amet consectetur adipisicing elit. Aliquam doloribus maxime at autem magnam numquam placeat nemo! Et aliquid quibusdam officiis necessitatibus fuga, eligendi facilis aut, laborum quod, esse repellat.

Footer Links

[Home](#)[Blog](#)[About](#)

visited, link, Home



Reach Router

- Keyboard accessible.
- Assistive devices announce changes.
- Cmd/Ctrl click opens in a new tab.
- Right click “open in new window” opens in a new window.
- Out-of-the-box focus management.

<https://reach.tech/router>



Announce page changes

- Announce page changes.
- Use refs to manage focus.
- If necessary, make items focusable by applying `tabindex="-1"`
- Check out **Reach Router**.

```
class Blog extends React.Component {
  constructor(props) {
    super(props);
    this.heading = React.createRef();
  }

  componentDidMount(){
    this.heading.current.focus();
    document.title = "Blog";
  }

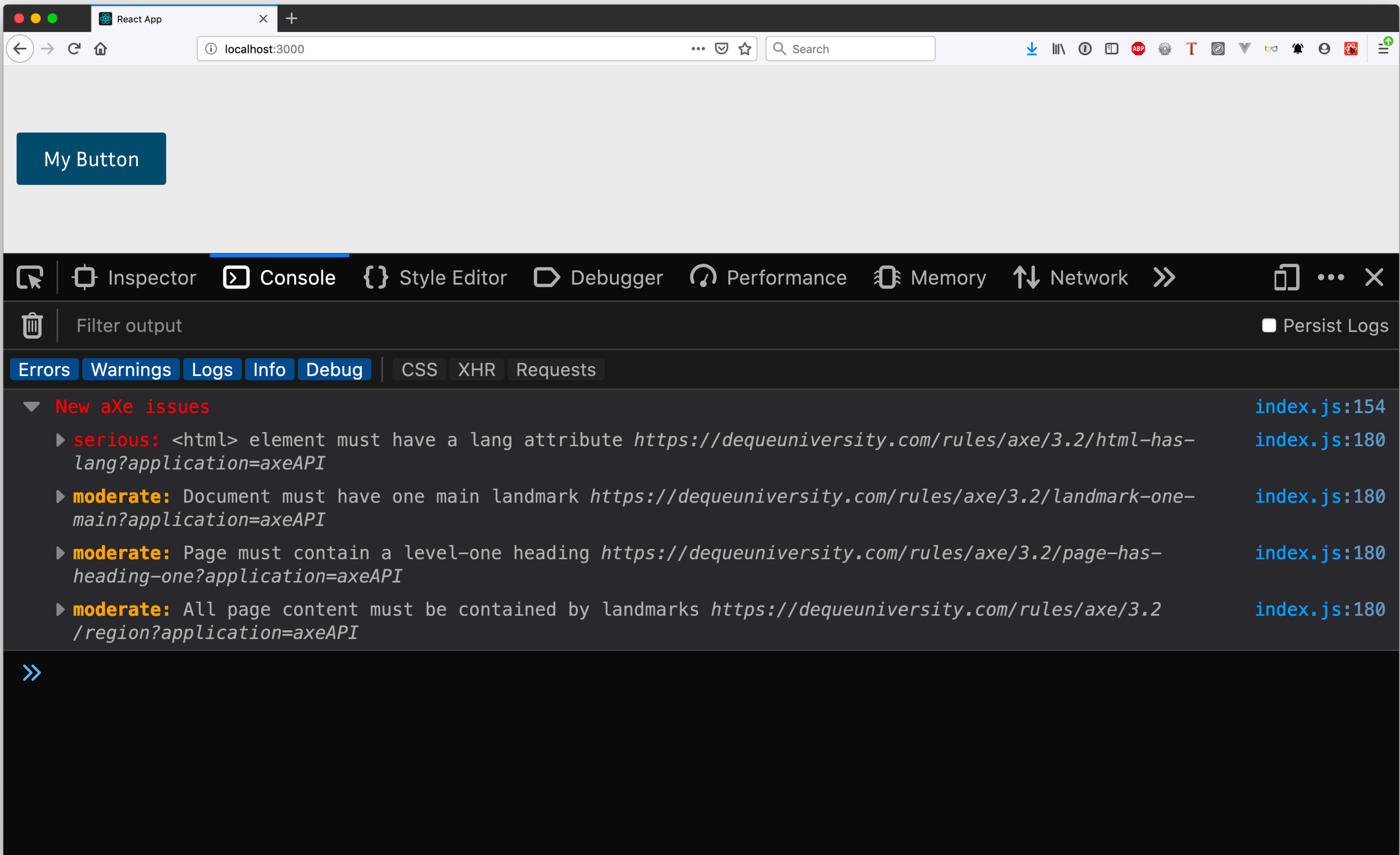
  render() {
    return (
      <section aria-labelledby="pageTitle"
        tabIndex="-1"
        ref= { this.heading }>
        <h2 id="pageTitle">Blog</h2><p>My blog...</p>
      </section>
    )
  }
}
```


a11y tip #8

Test your React
code automatically

a11y tip #5

```
var React = require('react');  
var ReactDOM = require('react-dom');  
  
if (process.env.NODE_ENV === 'production') {  
  var axe = require('react-axe');  
  axe(React, ReactDOM, 1000);  
}
```



Test your React code automatically

- Automatic tests help you notice low hanging fruits.
- Automatic testing is only the first step. Manual testing is necessary.
- Check out **React-axe** and **eslint-plugin-jsx-a11y** on Github.

```
var React = require('react');
var ReactDOM = require('react-dom');

if (process.env.NODE_ENV !== 'production') {
  var axe = require('react-axe');
  axe(React, ReactDOM, 1000);
}
```



💖 **Kiitos!** 💖

bit.ly/react-tips