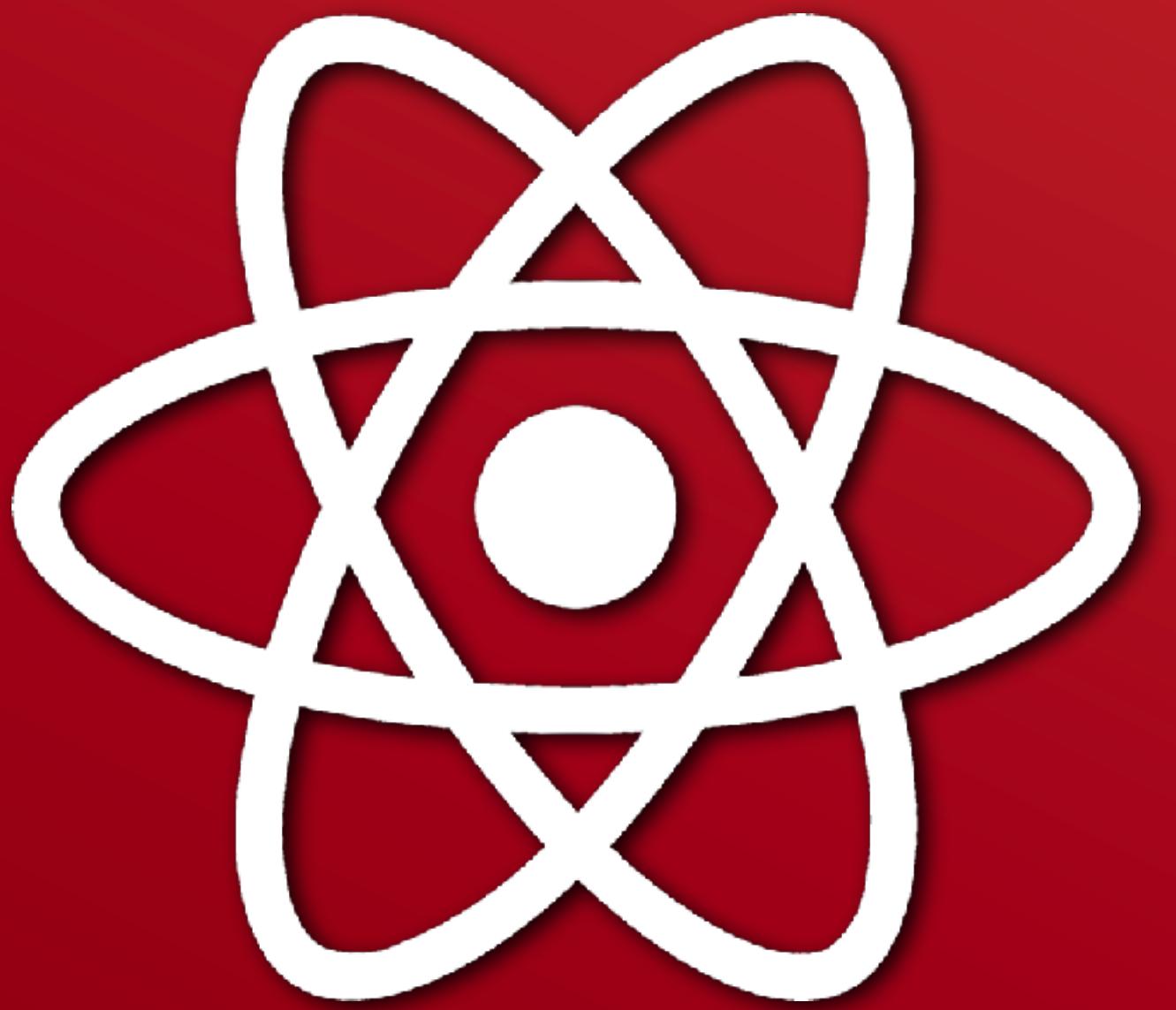


# EXTENDING HTML WITH WEB COMPONENTS

# DOBRÉ ODPOLEDNE!



# COMPONENT

reusable isolated part of UI with a single responsibility

# WEB COMPONENT



custom element

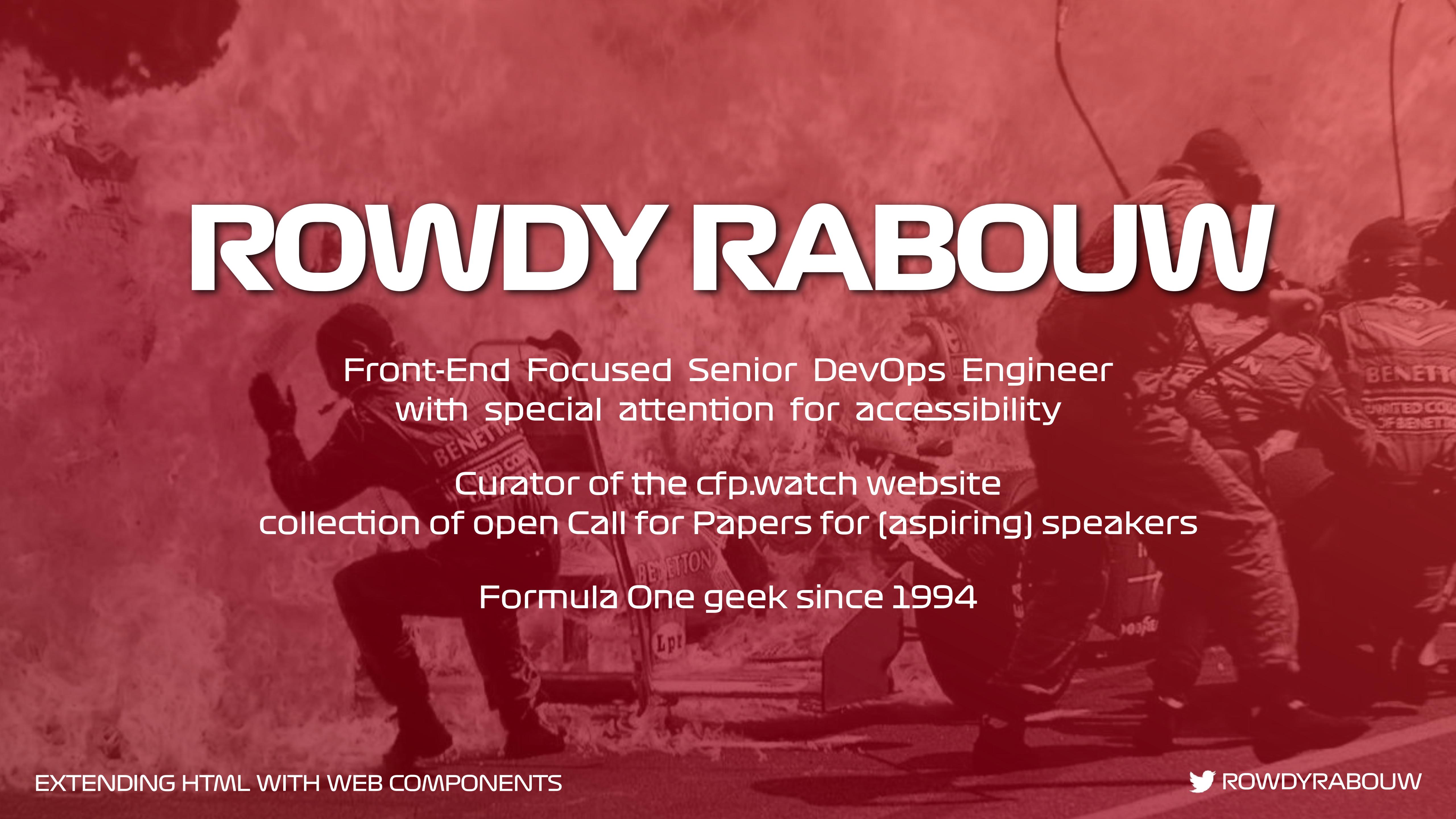


template



shadow DOM

# ROWDY RABOUW



Front-End Focused Senior DevOps Engineer  
with special attention for accessibility

Curator of the cfp.watch website  
collection of open Call for Papers for (aspiring) speakers

Formula One geek since 1994



Formula One Xplorer >> Current features OK

## WELCOME !

Good afternoon Formula One fan and welcome to the Formula One Xplorer website presented by Autodromo Formula Uno.

We are proud to present you the first **free** Formula One Windows program with regular updates in the world !

Are we mad ? No way ! We would like to share our growing collection of F1 data with you to promote one of the most exciting and hightech sports in the world.

## WHAT IS FORMULA ONE XPLORER ?

Formula One Xplorer (also known as FOX) is, as said above, a freeware Windows program. This means that it is totally free to use and share it with your friends and family.

FOX offers you the latest statistical information about teams and drivers, test results, Grand Prix results, various other statistics and charts ([a full features list](#) is available).

Data updates are available on Tuesdays after a Grand Prix week-end.  
These updates can be downloaded from this website or can be send to you [by email](#).

Program updates are released on a irregular basis.

## WHAT'S NEW ?

November 19, 2000 :

- FOX 2.0.0000.288 [bèta release](#) available

October 29, 2000 :

- FOX 1.2 [full version](#) released (including data version 2000.17)
- FOX 1.2 [complete download pack](#) released  
(including data version 2000.17 and Borland Database Engine 5.11)

SIRKUIT BALI PASTI TERWUJUD

<http://www.otosport-online.com>  
e-mail: otosport@goracing.net

NO. 15/I • SABTU 22 JULI 2000 • Rp 3.000

# otosport

**Prestone®**

RADIATOR SUPER COOLANT

- Mencegah Panas Berlebihan  
( Overheating )
- Meningkatkan Titik Didih Air

MICHAEL SCHUMACHER



-pukar untuk berlaga di sirkuit Monza?  
Jelaskan jawaban Anda ditinjau dari segi banyaknya bahan bakar yang diisi, pemakaian jenis ban, jenis front serta rear wing, dan karakteristik sirkuit Monza itu sendiri.

Essay dapat ditulis pada kertas yang terpisah.



Butuh waktu 1 tahun demi mewujudkan kecintaannya akan dunia F1. Bukan untuk berburu tandatangan atau foto bareng driver jet darat. Rowdy Rabouw, seorang fans asal Belanda, merilis peranti lunak yang mutakhir guna memenuhi hasratnya F1-nya.



"Satu tahun penuh saya berkutat dengan komputer untuk melahirkan software ini," tutur Rowdy. Namanya Formula One Xplorer 2000, disingkat jadi FOX 2000.

Fungsi peranti itu, menyimpan dan meng-update data statistik di Formula 1. Dari mulai biodata pembalap, tim, hasil kualifikasi setiap GP sampai hasil lomba dan hasil serta jadwal sesi testing. Lengkap dengan waktu, selisih dan per-

lehan poin. Masih termasuk klasemen pembalap dan konstruktur, plus jumlah lap terdepan. Tak lupa, jadwal tetap gelaran heboh itu. Gila memang.

Software gratis itu bisa di download via website miliknya,

No.	Driver	Team	Laps	Pos.	Speed
01	Michael Schumacher	Ferrari	116.476	200.115	
02	Mika Hakkinen	McLaren-Mercedes	116.087	0.213	199.959
03	Nick Heidfeld	Ponti-Peugeot	116.125	1.551	195.986
04	Ralf Schumacher	Williams-BMW	116.140	1.824	195.928
05	Eddie Irvine	Jaguar-Cosworth	116.225	1.761	195.611
06	Giancarlo Fisichella	Benetton-Peugeot	116.540	2.008	194.851
07	Johnny Herbert	Jaguar-Cosworth	116.585	2.112	194.732
08	Jean Alesi	Ponti-Peugeot	116.758	2.294	194.312
09	Pedro de la Rosa	Aston Martin	116.862	2.398	194.055
10	Jenson Button	Williams-BMW	116.968	2.495	193.783
11	Mika Salo	Sauber-Petronas	116.945	2.571	193.506
12	Alexander Wurz	Benetton-Peugeot	116.981	2.607	193.316
13	Jacques Villeneuve	BAR-Honda	116.130	2.696	193.208
14	Jos Verstappen	Aston Martin	116.550	3.076	192.377
15	Ricardo Zonta	BAR-Honda	116.727	3.253	191.959
16	Pedro Diniz	Sauber-Petronas	116.748	3.294	191.899
17	Gastón Mazzacane	Minardi-Pondretal	116.619	4.145	190.828
18	Mac Bonne	Minardi-Pondretal	121.180	4.725	182.470
19/1			121.027	5.353	182.023
David Coulthard	McLaren-Mercedes	no time			
Rubens Barrichello	Ferrari	no time			
Heinz-Harald Frentzen	Jordan-Mugen Honda	no time			
Jarno Trulli	Jordan-Mugen Honda	no time			

Tanda tangan:

Tanggal:



<http://fox.autodromo.com/>. Syaratnya, komputer sasaran harus punya Borland Database Engine (BDE) versi terakhir guna menyokong peranti asyik itu. Namun jangan khawatir. "Saya sudah memikirkan semuanya, termasuk menyediakan BDE gratis di situs yang sama," tutur pria asal Gouda, Belanda itu. "Install dulu BDE tadi, baru FOX-nya."

Namun kurang gila kalau ceritanya selesai sampai di situ. Selain membuat software tersebut, Rowdy juga menyediakan *data updating* gratis. Pasalnya, usai setiap GP, statistik pasti berubah. Hebatnya, data-data itu bisa diperoleh dari websitenya cuma dua hari pasca lomba. Jadi kalau GP Austria digelar pada 16 Juli, hari Selasa (18/7) sudah ada update-nya.

Lebih gila lagi, karena sang pembuat FOX itu masih belum puas jua. "Saya masih bekerja untuk membuat revisi terbarunya yang bisa dibaca dalam bahasa lain, mungkin juga bahasa Indonesia," sambungnya lagi. "Semua ini demi kecintaan saya pada Formula 1," katanya pada *otosport*.

Sudah lengkap, gratis pula. Padahal, ia punya pilihan untuk menjual software hebat macam itu. Nah, ini dia fans sejati! ■ **Arya A. Sadhana**

# Prancis Terancam Punah

Grand Prix F1 2000  
... menampilkan se-

Kolom Gérard  
'Jabby' Crombac

# TOMÁŠ ENGE



# SCHEDULE

- Custom Elements
- Templates
- Shadow DOM
- Properties & Attributes
- Default & Named Slots
- Lifecycle Events
- Dispatching Events
- Web Components & JS Frameworks

# CUSTOM ELEMENT

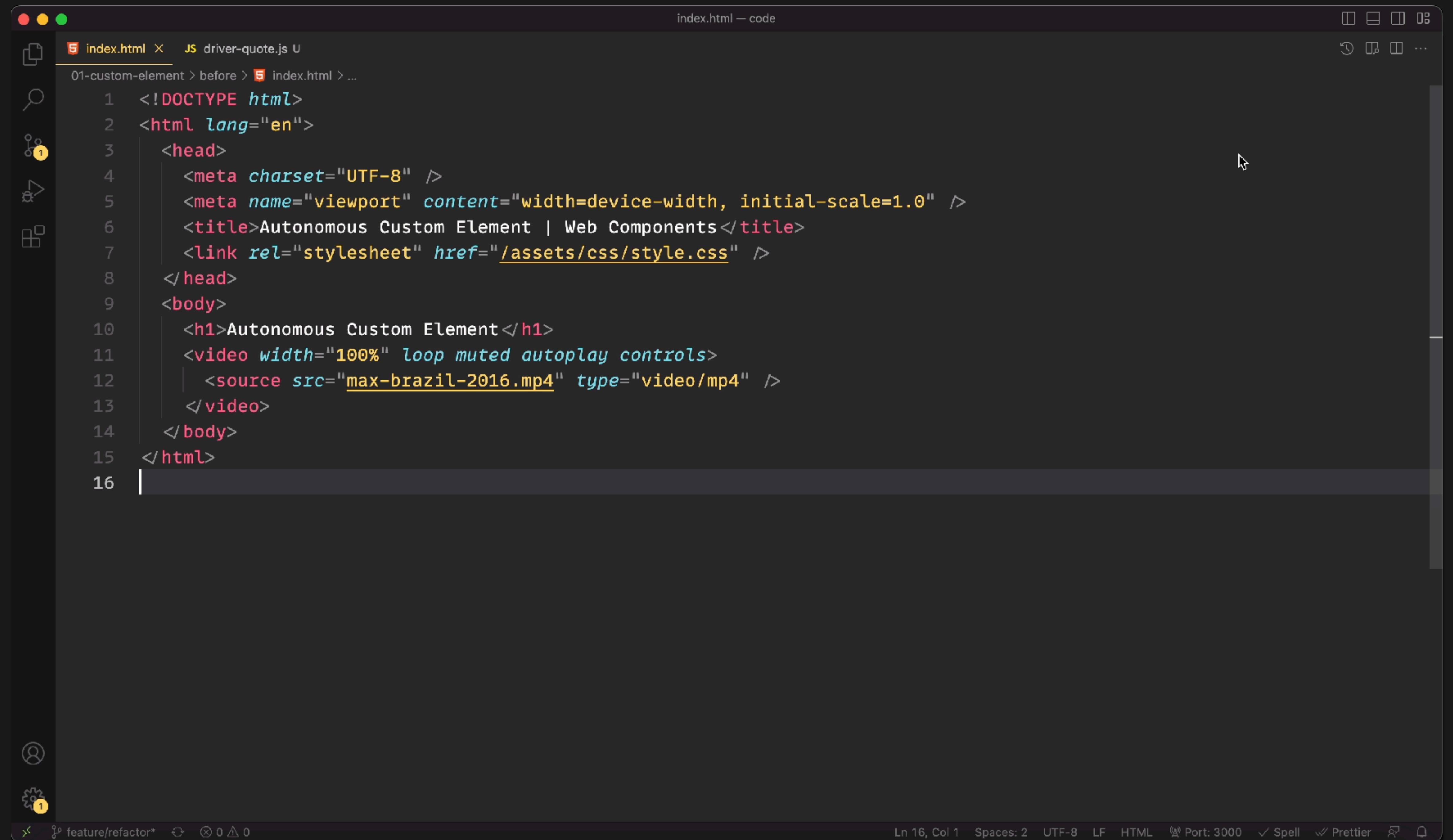
define your own HTML element and attach behaviour

or

alter a native HTML element and attach behaviour

# AUTONOMOUS CUSTOM ELEMENT

define your own HTML element



```
class DriverQuote extends HTMLElement {  
  constructor() {  
    super();  
    this.innerHTML = `<blockquote><p>Because I'm Dutch</p>  
      <cite>Max Verstappen</cite></blockquote>`;  
  }  
}  
  
customElements.define("driver-quote", DriverQuote);
```

- ES6 class definition → modern browsers
- Extend HTMLElement → call its constructor with super()
- Set the innerHTML of our component
- Register the component to the Document Object Model (DOM)

```
<driver-quote></driver-quote>  
<script src="driver-quote.js"></script>
```

- Insert the tag (dash required)
- Include the Custom Component JavaScript

# CUSTOMISED NATIVE ELEMENT

alter a native HTML element

index.html – code

index.html X JS external-link.js U

02-native-element > before > index.html > ...

```
1  <!DOCTYPE html>
2  <html lang="en">
3      <head>
4          <meta charset="UTF-8" />
5          <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6          <title>Customised Native Element | Web Components</title>
7          <link rel="stylesheet" href="/assets/css/style.css" />
8      </head>
9      <body>
10         <h1>Customised Native Element</h1>
11         <sub>run this demo in a browser, not in VS Code live preview</sub>
12         <ul>
13             <li><a href="https://www.sauber-group.com/motorsport/formula-1/">Alfa Romeo</a></li>
14             <li><a href="https://scuderia.alphatauri.com/en/">AlphaTauri</a></li>
15             <li><a href="https://www.alpine-cars.co.uk/formula-1.html">Alpine</a></li>
16             <li><a href="https://www.astonmartinf1.com/en-GB/">Aston Martin</a></li>
17             <li><a href="https://www.ferrari.com/en-EN/formula1">Ferrari</a></li>
18             <li><a href="https://www.haasf1team.com">Haas</a></li>
19             <li><a href="https://www.mclaren.com/racing/formula-1/">McLaren</a></li>
20             <li><a href="https://www.mercedesamgf1.com">Mercedes</a></li>
21             <li><a href="https://www.redbullracing.com">Red Bull</a></li>
22             <li><a href="https://www.williamsf1.com">Williams</a></li>
23         </ul>
24     </body>
25 </html>
```

feature/refactor\* ⌂ ⌂ 0 △ 0 ⌂ 2

Ln 26, Col 1 Spaces: 2 UTF-8 LF HTML ⚙ Port: 3000 ⚙ 2 Spell ⚙ Prettier ⌂ ⌂

```
class ExternalLink extends HTMLAnchorElement {  
  
  constructor() {  
    ...  
  }  
  
  customElements.define("external-link", ExternalLink, { extends: "a" });
```

- Extend a specific HTMLElement
- Register the extend to the Document Object Model (DOM)

```
<li><a is="external-link" href="https://www.redbullracing.com/int-en">Red Bull</a></li>
```

- Add `is`-attribute with the tag name of your custom element



Webkit does not and will not ever support it.  
They want it removed from the spec.

However, a polyfill is available:  
<https://github.com/ungap/custom-elements>

# TEMPLATES

```
class DriverQuote extends HTMLElement {
  constructor() {
    super();
    this.innerHTML = `<blockquote><p>Because I'm Dutch</p>
      <cite>Max Verstappen</cite></blockquote>`;
  }
}

customElements.define("driver-quote", DriverQuote);
```

```
const template = document.createElement("template");
template.innerHTML = `
  <blockquote><p>Because I'm Dutch</p><ccite>Max Verstappen</ccite></blockquote>`;

class DriverQuote extends HTMLElement {
  constructor() {
    super();
    this.appendChild(template.content.cloneNode(true));
  }
}

customElements.define("driver-quote", DriverQuote);
```

# STYLING

index.html – code

index.html X JS driver-quote.js

04-global-css > before > index.html > html > head > link

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Global CSS | Web Components</title>
7     <link rel="stylesheet" href="/assets/css/style.css" />
8   </head>
9   <body>
10    <h1>Global CSS</h1>
11    <blockquote>
12      <p>If you no longer go for a gap that exists, you're no longer a racing driver</p>
13      <cite>Ayrton Senna</cite>
14    </blockquote>
15    <driver-quote></driver-quote>
16    <script src="driver-quote.js"></script>
17  </body>
18</html>
```

Global CSS | Web Components X

http://127.0.0.1:3000/04-global-css/before/index.html

# Global CSS

“*If you no longer go for a gap that exists, you're no longer a racing driver*”

– Ayrton Senna

“*Because I'm Dutch*”

– Max Verstappen

feature/refactor ⌘ ⌘ 0 △ 0 ⌂ 1

Ln 7, Col 59 Spaces: 2 UTF-8 LF HTML Port: 3000 1 Spell Prettier

# SHADOW DOM

CSS Encapsulation

index.html – code

index.html x JS driver-quote.js

05-shadow-dom > before > index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Shadow DOM | Web Components</title>
7     <link rel="stylesheet" href="/assets/css/style.css" />
8     <style>
9      blockquote {
10         color: #24316e !important;
11       }
12     </style>
13   </head>
14   <body>
15     <h1>Shadow DOM</h1>
16     <blockquote>
17       <p>If you no longer go for a gap that exists, you're no longer a racing driver</p>
18       <cite>Ayrton Senna</cite>
19     </blockquote>
20     <driver-quote></driver-quote>
21     <script src="driver-quote.js"></script>
22   </body>
23 </html>
```

Shadow DOM | Web Components x

http://127.0.0.1:3000/05-shadow-dom/before/index.html

# Shadow DOM

“*If you no longer go for a gap that exists, you're no longer a racing driver*”

– Ayrton Senna

“*Because I'm Dutch*”

– Max Verstappen

feature/refactor\* ⌂ ⌂ 0 △ 0 ⌂ 1

Ln 24, Col 1 Spaces: 2 UTF-8 LF HTML Port: 3000 1 Spell Prettier ⌂ ⌂



Shadow DOM | Web Component X

Shadow DOM | Web Component X



127.0.0.1:3000/05-shadow-dom/before/index.html



# Shadow DOM

“*If you no longer go for a gap that exists, you’re no longer a racing driver*”

– Ayrton Senna

“*Because I’m Dutch*”

– Max Verstappen

# CSS VARIABLES

officially: CSS custom properties

The screenshot shows a code editor interface with two tabs: `index.html` and `driver-quote.js`. The `index.html` tab displays the following HTML code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>CSS Variables | Web Components</title>
7     <link rel="stylesheet" href="/assets/css/style.css" />
8     <style>
9      blockquote {
10         color: #24316e !important;
11       }
12     </style>
13   </head>
14   <body>
15     <h1>CSS Variables</h1>
16     <blockquote>
17       <p>If you no longer go for a gap that exists, you're no longer a racing driver</p>
18       <cite>Ayrton Senna</cite>
19     </blockquote>
20     <driver-quote></driver-quote>
21     <script src="driver-quote.js"></script>
22   </body>
23 </html>
```

The `driver-quote.js` tab contains the following JavaScript code:

```
<!-- This file is generated by Prettier -->
```

The browser preview on the right shows the rendered HTML with the title "CSS Variables" and a quote from Ayrton Senna: "If you no longer go for a gap that exists, you're no longer a racing driver". Below it is another quote from Max Verstappen: "Because I'm Dutch".

# CSS THEMING

index.html — code

index.html × JS driver-quote.js JS boss-quote.js

07-theme > before > index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>CSS Theming | Web Components</title>
7     <link rel="stylesheet" href="/assets/css/style.css" />
8   </head>
9   <body>
10    <h1>CSS Theming</h1>
11    <driver-quote></driver-quote>
12    <boss-quote></boss-quote>
13    <script type="module" src="index.js"></script>
14  </body>
15 </html>
```

CSS Theming | Web Components

http://127.0.0.1:3000/07-theme/before/index.h

# CSS Theming

“Because I'm Dutch.”

– Max Verstappen

“Mikey, that is so not right. That is so not right.”

– Toto Wolff

feature/refactor ⌘ 0 ▲ 0 ⌂ 1

Ln 16, Col 1 Spaces: 2 UTF-8 LF HTML Port: 3000 ✓ Spell ✎ Prettier

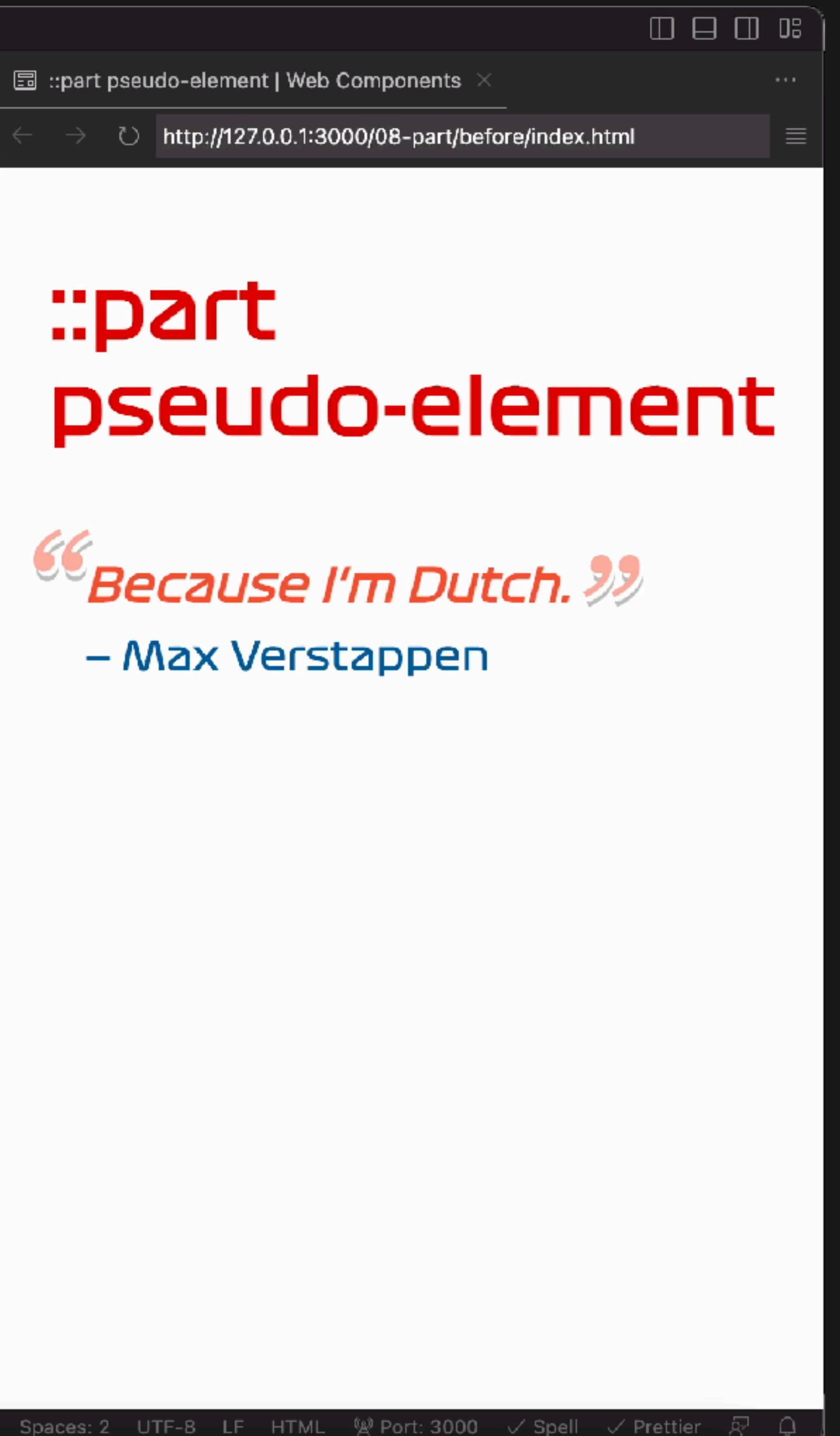
# BETTER CSS THEMING

::part pseudo-element

The screenshot shows a dark-themed code editor interface with the following details:

- Top Bar:** Shows file tabs for "index.html" and "driver-quote.js", and a status bar indicating "index.html — code".
- Sidebar:** On the left, there are several icons with yellow notification bubbles: a clipboard (1), a magnifying glass (1), a circular progress bar (1), a double arrow (1), and a gear (1).
- Code Area:** The main area displays the content of "index.html".

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>::part pseudo-element | Web Components</title>
7      <link rel="stylesheet" href="/assets/css/style.css" />
8  </head>
9  <body>
10     <h1>::part <span>pseudo-element</span></h1>
11     <driver-quote></driver-quote>
12     <script type="module" src="index.js"></script>
13 </body>
14 </html>
15 |
```
- Status Bar:** At the bottom, it shows "feature/refactor\*" and "Ln 15, Col 1".



# PROPERTIES

```
class DriverQuote extends HTMLElement {  
  
    set quote(value) {  
        this._quote = value;  
        this.quoteElement.innerHTML = this._quote;  
    }  
  
    get quote() { return this._quote; }  
  
    set driver(value) {  
        this._driver = value;  
        this.driverElement.innerHTML = this._driver;  
    }  
  
    get driver() { return this._driver; }  
  
    constructor() {  
        super();  
        this.attachShadow({ mode: "open" });  
        this.shadowRoot.appendChild(template.content.cloneNode(true));  
        this.quoteElement = this.shadowRoot.querySelector("p");  
        this.driverElement = this.shadowRoot.querySelector("cite");  
    }  
}  
  
customElements.define("driver-quote", DriverQuote);
```

```
class DriverQuote extends HTMLElement {  
  
    set quote(value) {  
        this._quote = value;  
        this.quoteElement.innerHTML = this._quote;  
    }  
  
    get quote() { return this._quote; }  
  
    set driver(value) {  
        this._driver = value;  
        this.driverElement.innerHTML = this._driver;  
    }  
  
    get driver() { return this._driver; }  
  
    constructor() {  
        super();  
        this.attachShadow({ mode: "open" });  
        this.shadowRoot.appendChild(template.content.cloneNode(true));  
        this.quoteElement = this.shadowRoot.querySelector("p");  
        this.driverElement = this.shadowRoot.querySelector("cite");  
    }  
}  
  
customElements.define("driver-quote", DriverQuote);
```

index.html — code

index.html X JS driver-quote.js JS index.js

09-properties > before > index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Properties | Web Components</title>
7     <link rel="stylesheet" href="/assets/css/style.css" />
8   </head>
9   <body>
10    <h1>Properties</h1>
11    <driver-quote></driver-quote>
12    <script type="module" src="index.js"></script>
13  </body>
14 </html>
15 
```

Properties

“ ”

feature/refactor ⌂ ⌂ 0 △ 0

Ln 15, Col 1 Spaces: 2 UTF-8 LF HTML Port: 3000 ✓ Spell ✎ Prettier

# ATTRIBUTES

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Attributes | Web Components</title>
    <link rel="stylesheet" href="/assets/css/style.css" />
  </head>
  <body>
    <h1>Attributes</h1>

    <driver-quote data-quote="Because I'm Dutch"
      data-driver="Max Verstappen"></driver-quote>

    <script type="module" src="index.js"></script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Attributes | Web Components</title>
    <link rel="stylesheet" href="/assets/css/style.css" />
  </head>
  <body>
    <h1>Attributes</h1>

    <driver-quote data-quote="Because I'm Dutch"
      data-driver="Max Verstappen"></driver-quote>

    <script type="module" src="index.js"></script>
  </body>
</html>
```

```
class DriverQuote extends HTMLElement {
  static get observedAttributes() {
    return ["data-quote", "data-driver"];
  }

  set quote(value) { ... }
  get quote() { ... }
  set driver(value) { ... }
  get driver() { ... }
  constructor() { ... }

  attributeChangedCallback(attrName, oldValue, newValue) {
    switch (attrName) {
      case "data-quote":
        this.quote = newValue;
        break;
      case "data-driver":
        this.driver = newValue;
        break;
    }
  }
}

customElements.define("driver-quote", DriverQuote);
```

```
class DriverQuote extends HTMLElement {  
    static get observedAttributes() {  
        return ["data-quote", "data-driver"];  
    }  
  
    set quote(value) { ... }  
    get quote() { ... }  
    set driver(value) { ... }  
    get driver() { ... }  
    constructor() { ... }  
  
    attributeChangedCallback(attrName, oldValue, newValue) {  
        switch (attrName) {  
            case "data-quote":  
                this.quote = newValue;  
                break;  
            case "data-driver":  
                this.driver = newValue;  
                break;  
        }  
    }  
}  
  
customElements.define("driver-quote", DriverQuote);
```

index.html — code

index.html x JS index.js

10-attributes > before > index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Attributes | Web Components</title>
7     <link rel="stylesheet" href="/assets/css/style.css" />
8   </head>
9   <body>
10    <h1>Attributes</h1>
11    <driver-quote></driver-quote>
12    <script type="module" src="index.js"></script>
13  </body>
14 </html>
15 |
```

Attributes | Web Components x

http://127.0.0.1:3000/10-attributes/before/index.h

# Attributes

“ ”

feature/refactor ⌂ ⌂ 0 △ 0

Ln 15, Col 1 Spaces: 2 UTF-8 LF HTML Port: 3000 ✓ Spell ✓ Prettier ⌂

A black and white photograph of a Formula 1 race track in Monaco. In the foreground, a large white flag is being held by a person in a racing helmet. In the background, a Formula 1 car is driving on the track, and there are spectators in the stands and a city skyline in the distance.

# WARNING!

attributes can only accept string values  
properties accept any kind of values, like Objects or Arrays

### 3.2.6.6 Embedding custom non-visible data with the **data-\*** attributes §

A **custom data attribute** is an attribute in no namespace whose name starts with the string "**data-**", has at least one character after the hyphen, is [XML-compatible](#), and contains no [ASCII upper alphas](#).

#### Note

*All attribute names on [HTML elements](#) in [HTML documents](#) get ASCII-lowercased automatically, so the restriction on ASCII uppercase letters doesn't affect such documents.*

[Custom data attributes](#) are intended to store custom data, state, annotations, and similar, private to the page or application, for which there are no more appropriate attributes or elements.

# SLOTS

```
<driver-quote data-quote="Because I'm Dutch" data-driver="Max Verstappen"></driver-quote>
```

<driver-quote>Because I'm Dutch</driver-quote>

```
template.innerHTML = `<blockquote><p></p><cite></cite></blockquote>`;
```

```
template.innerHTML = `<blockquote><p><slot></slot></p><cite></cite></blockquote>`;
```

```
template.innerHTML = `<blockquote><p><slot></slot></p><cite></cite></blockquote>` ;
```

index.html — code

index.html × JS driver-quote.js

11-slots > before > index.html > ...

```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8" />
5     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6     <title>Slots | Web Components</title>
7     <link rel="stylesheet" href="/assets/css/style.css" />
8   </head>
9   <body>
10    <h1>Slots</h1>
11    <driver-quote></driver-quote>
12    <script type="module" src="index.js"></script>
13  </body>
14 </html>
15 |
```

Slots | Web Components

http://127.0.0.1:3000/11-slots/before/index.html

The screenshot shows a dark-themed code editor with two tabs open: 'index.html' and 'JS driver-quote.js'. The 'index.html' tab contains the provided HTML code. The browser preview on the right shows the word 'Slots' in a large red font, with two orange speech marks ('“”') positioned below it. The status bar at the bottom indicates the file is at line 15, column 1, with spaces: 2, encoding: UTF-8, line feed: LF, and HTML mode.

# NAMED SLOTS

<driver-quote>Because I'm Dutch</driver-quote>

```
<driver-quote data-image="max-verstappen.jpg">
  Because I'm Dutch
  <span slot="driver">Max Verstappen</span>
</driver-quote>
```

```
template.innerHTML = `<blockquote><p><slot></slot></p><cite></cite></blockquote>`;
```

```
template.innerHTML = `<div class="card"><img alt=""><cite><slot name="driver"></slot></cite><blockquote><p><slot></slot></p></blockquote></div>`;
```

```
template.innerHTML = `<div class="card"><img alt=""><cite><slot name="driver"></slot></cite><blockquote><p><slot></slot></p></blockquote></div>`;
```

```
class DriverQuote extends HTMLElement {
  static get observedAttributes() {
    return ["data-image"];
  }

  set image(value) {
    this._image = value;
    this.imgElement.src = this._image;
  }

  get image() {
    return this._image;
  }

  constructor() {
    ...
    this.imgElement = this.shadowRoot.querySelector("img");
  }

  attributeChangedCallback(attrName, oldValue, newValue) {
    if (attrName === "data-image") { this.image = newValue; }
  }
}

customElements.define("driver-quote", DriverQuote);
```

```
class DriverQuote extends HTMLElement {  
    static get observedAttributes() {  
        return ["data-image"];  
    }  
  
    set image(value) {  
        this._image = value;  
        this.imgElement.src = this._image;  
    }  
  
    get image() {  
        return this._image;  
    }  
  
    constructor() {  
        ...  
        this.imgElement = this.shadowRoot.querySelector("img");  
    }  
  
    attributeChangedCallback(attrName, oldValue, newValue) {  
        if (attrName === "data-image") { this.image = newValue; }  
    }  
}  
  
customElements.define("driver-quote", DriverQuote);
```

snippet.html — code

snippet.html X driver-quote.js index.html

12-named-slots > before > snippet.html > style

```
1 <style>
2   @import "quote.css";
3 </style>
4 <style>
5   .card {
6     border-radius: 1rem;
7     background: #bd0019;
8     border: 1px solid #000000;
9     display: flex;
10    flex-direction: column;
11  }
12  img {
13    max-width: 100%;
14    border-top-left-radius: 1rem;
15    border-top-right-radius: 1rem;
16  }
17  cite {
18    font-size: 2rem;
19    margin: 0rem;
20    text-align: center;
21    color: #fff;
22    background-color: #000;
23    padding: 1rem 0;
24  }
25  blockquote {
26    color: #fff;
27    margin: 1rem 0 2rem 0;
28    width: 100%;
29    text-align: center;
```

Named Slots | Web Components

http://127.0.0.1:3000/12-named-slots/before/index.html

# Named Slots

“Because I'm Dutch”

—

feature/refactor\* 0↓ 1↑ ⌂ 0 ▲ 0

Ln 1, Col 8 Spaces: 2 UTF-8 LF HTML Port: 3000 ✓ Spell ✎ Prettier

# LIFECYCLE EVENTS

## **attributeChangedCallback(attrName, oldVal, newVal)**

invoked each time one of the **attributes** is added, removed, or changed

## **static get observedAttributes() { return ["attribute"]; }**

array of attributes to observe

## **connectedCallback()**

invoked each time the custom element is added to the DOM

## **disconnectedCallback()**

invoked each time the custom element is removed from the DOM

# **adoptedCallback()**

invoked each time the custom element is moved to a new document

Lifecycle | Web Components

127.0.0.1:3000/13-lifecycle/index.html

# Lifecycle

run this demo in a browser, not in VS Code live preview

Add/remove component

Example Component

Example Component in iFrame

Adopt component

Elements Console Sources Network > Filter All levels ▾ No Issues

# EVENTS

```
this.dispatchEvent(new CustomEvent("event-name", { detail: data }));
```

```
component.addEventListener("event-name", (e) => {
  console.log(e.detail);
});
```

you can also use `new Event` if you don't need to send details

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Events | Web Components</title>
    <link rel="stylesheet" href="/assets/css/style.css" />
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.8.0/dist/leaflet.css" />
    <script src="https://unpkg.com/leaflet@1.8.0/dist/leaflet.js"></script>
  </head>
  <body>
    <h1>Events</h1>
    <sub>run this demo in a browser, not in VS Code live preview</sub>

    <random-circuit></random-circuit>

    <div id="result"></div>
    <div id="map"></div>

    <script type="module" src="index.js"></script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Events | Web Components</title>
    <link rel="stylesheet" href="/assets/css/style.css" />
    <link rel="stylesheet" href="https://unpkg.com/leaflet@1.8.0/dist/leaflet.css" />
    <script src="https://unpkg.com/leaflet@1.8.0/dist/leaflet.js"></script>
  </head>
  <body>
    <h1>Events</h1>
    <sub>run this demo in a browser, not in VS Code live preview</sub>

    <random-circuit></random-circuit>

    <div id="result"></div>
    <div id="map"></div>

    <script type="module" src="index.js"></script>
  </body>
</html>
```

```
const template = document.createElement("template");
template.innerHTML = `<button>Dispatch event</button>`;

class RandomCircuit extends HTMLElement {
    static _circuits;

    constructor() {
        super();
        fetch("http://ergast.com/api/f1/drivers/max_verstappen/circuits.json?limit=100")
            .then((res) => res.json())
            .then((data) => {
                this._circuits = data.MRData.CircuitTable.Circuits;
            });

        this.attachShadow({ mode: "open" });
        this.shadowRoot.appendChild(template.content.cloneNode(true));
        this.shadowRoot.querySelector("button").addEventListener("click", ()=>this.dispatchClick());
    }

    dispatchClick() {
        const randomCircuit = this._circuits[Math.floor(Math.random() * this._circuits.length)];
        this.dispatchEvent(new CustomEvent("getRandomCircuit", { detail: randomCircuit }));
    }
}

customElements.define("random-circuit", RandomCircuit);
```

```
const template = document.createElement("template");
template.innerHTML = `<button>Dispatch event</button>`;

class RandomCircuit extends HTMLElement {
  static _circuits;

  constructor() {
    super();
    fetch("http://ergast.com/api/f1/drivers/max_verstappen/circuits.json?limit=100")
      .then((res) => res.json())
      .then((data) => {
        this._circuits = data.MRData.CircuitTable.Circuits;
      });

    this.attachShadow({ mode: "open" });
    this.shadowRoot.appendChild(template.content.cloneNode(true));
    this.shadowRoot.querySelector("button").addEventListener("click", ()=>this.dispatchClick());
  }

  dispatchClick() {
    const randomCircuit = this._circuits[Math.floor(Math.random() * this._circuits.length)];
    this.dispatchEvent(new CustomEvent("getRandomCircuit", { detail: randomCircuit }));
  }
}

customElements.define("random-circuit", RandomCircuit);
```

```
import "./random-circuit.js";

const component = document.querySelector("random-circuit");
const result = document.querySelector("#result");
const circuitMap = document.querySelector("#map");

const map = L.map("map").setView([-23.7036, -46.6997], 14);
L.tileLayer("https://api.mapbox.com/...", {
  attribution: "Map data &copy; ...<a>",
  maxZoom: 18,
  id: "mapbox/streets-v11",
  tileSize: 512,
  zoomOffset: -1,
  accessToken: "pk.eyJ1Ijoicm93...",
}) .addTo(map);

component.addEventListener("getRandomCircuit", (e) => {
  const circuit = e.detail;
  result.innerHTML =
    `

## ${circuit.circuitName}</h2> ${circuit.Location.locality}, ${circuit.Location.country} `; map.setView([circuit.Location.lat, circuit.Location.long], 14); circuitMap.style.visibility = "visible"; });


```

```
import "./random-circuit.js";

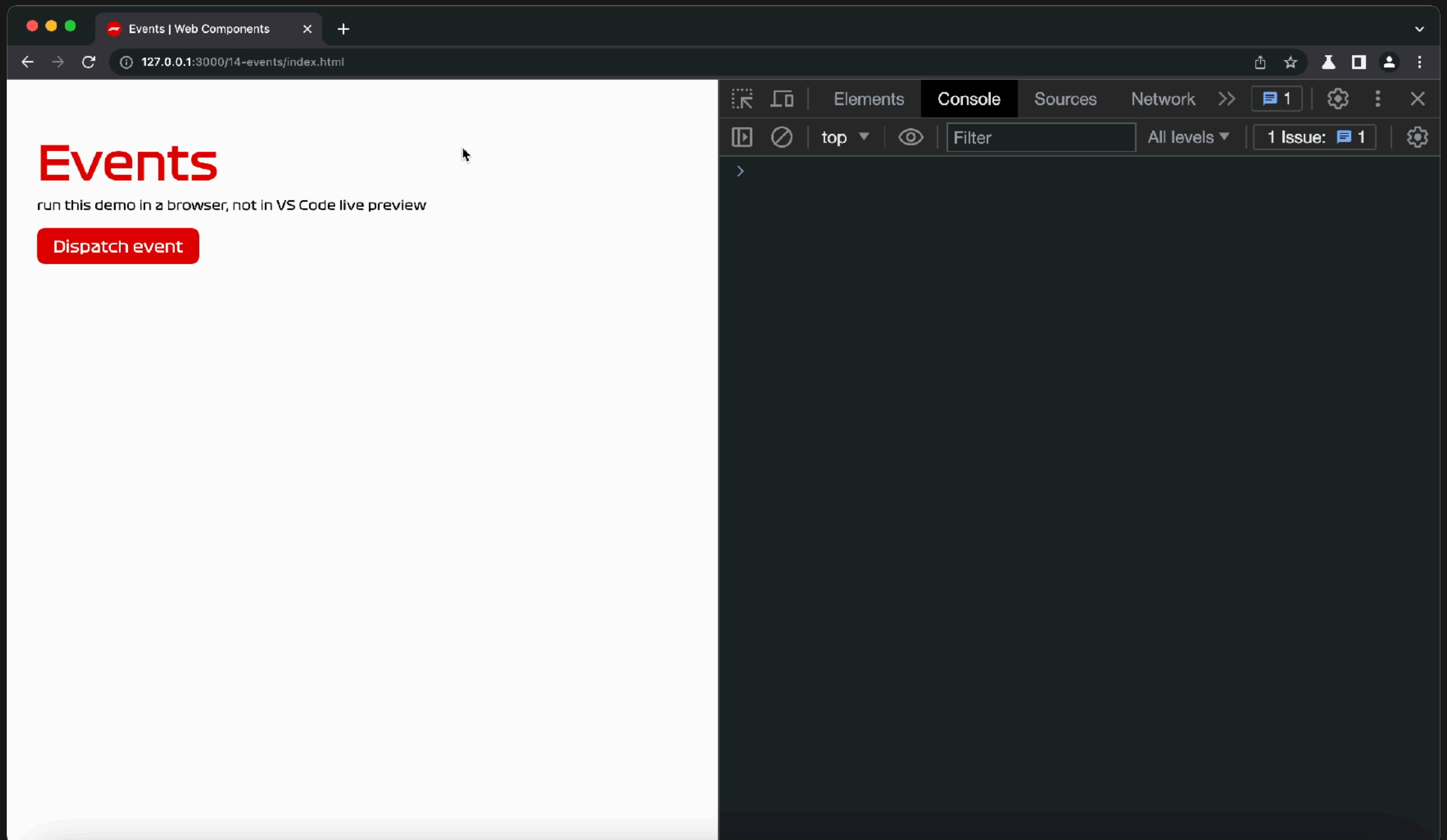
const component = document.querySelector("random-circuit");
const result = document.querySelector("#result");
const circuitMap = document.querySelector("#map");

const map = L.map("map").setView([-23.7036, -46.6997], 14);
L.tileLayer("https://api.mapbox.com/...", {
  attribution: "Map data &copy; ...<a>",
  maxZoom: 18,
  id: "mapbox/streets-v11",
  tileSize: 512,
  zoomOffset: -1,
  accessToken: "pk.eyJ1Ijoicm93 ...",
}) .addTo(map);

component.addEventListener("getRandomCircuit", (e) => {
  const circuit = e.detail;
  result.innerHTML =
    `

## ${circuit.circuitName}</h2> ${circuit.Location.locality}, ${circuit.Location.country} `; map.setView([circuit.Location.lat, circuit.Location.long], 14); circuitMap.style.visibility = "visible"; });


```



# EVENTS BETWEEN COMPONENTS

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Component Interaction | Web Components</title>
    <link rel="stylesheet" href="/assets/css/style.css" />
  </head>
  <body>
    <h1>Component Interaction</h1>

    <driver-list></driver-list>

    <driver-detail></driver-detail>

      <script type="module" src="index.js"></script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Component Interaction | Web Components</title>
    <link rel="stylesheet" href="/assets/css/style.css" />
  </head>
  <body>
    <h1>Component Interaction</h1>

    <driver-list></driver-list>

    <driver-detail></driver-detail>

    <script type="module" src="index.js"></script>
  </body>
</html>
```

```
const template = document.createElement("template");
template.innerHTML = `<select></select>`;

class DriverList extends HTMLElement {
  get drivers() {
    return this._drivers;
  }

  set drivers(value) {
    this._drivers = value;
    this.render();
  }

  constructor() {
    super();
    this.attachShadow({ mode: "open" });
    this.shadowRoot.appendChild(template.content.cloneNode(true));
    this.driverListElement = this.shadowRoot.querySelector("select");
  }
}
```

```
const template = document.createElement("template");
template.innerHTML = `<select></select>`;

class DriverList extends HTMLElement {
  get drivers() {
    return this._drivers;
  }

  set drivers(value) {
    this._drivers = value;
    this.render();
  }

  constructor() {
    super();
    this.attachShadow({ mode: "open" });
    this.shadowRoot.appendChild(template.content.cloneNode(true));
    this.driverListElement = this.shadowRoot.querySelector("select");
  }
}
```

```
render() {
  this._drivers.forEach((driver) => {
    const option = document.createElement("option");
    option.value = driver.driverId;
    option.textContent = `${driver.familyName}, ${driver.givenName}`;
    this.driverListElement.appendChild(option);
  });

  this.driverListElement.addEventListener("change", () => {
    const driver = this._drivers.find(
      (driver) => driver.driverId === this.driverListElement.value);
    this.dispatchEvent(new CustomEvent("selectDriver", { detail: driver }));
  });
}

customElements.define("driver-list", DriverList);
```

```
render() {
  this._drivers.forEach((driver) => {
    const option = document.createElement("option");
    option.value = driver.driverId;
    option.textContent = `${driver.familyName}, ${driver.givenName}`;
    this.driverListElement.appendChild(option);
  });

  this.driverListElement.addEventListener("change", () => {
    const driver = this._drivers.find(
      (driver) => driver.driverId === this.driverListElement.value);
    this.dispatchEvent(new CustomEvent("selectDriver", { detail: driver }));
  });
}

customElements.define("driver-list", DriverList);
```

```
class DriverDetail extends HTMLElement {  
  
  get driver() {  
    return this._driver;  
  }  
  
  set driver(value) {  
    this._driver = value;  
    this.render();  
  }  
  
  constructor() {  
    super();  
    this.attachShadow({ mode: "open" });  
  }  
  
  readableDate(dateString) {  
    const options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };  
    return new Date(dateString).toLocaleDateString("en-GB", options);  
  }  
}
```

```
class DriverDetail extends HTMLElement {  
  
  get driver() {  
    return this._driver;  
  }  
  
  set driver(value) {  
    this._driver = value;  
    this.render();  
  }  
  
  constructor() {  
    super();  
    this.attachShadow({ mode: "open" });  
  }  
  
  readableDate(dateString) {  
    const options = { weekday: 'long', year: 'numeric', month: 'long', day: 'numeric' };  
    return new Date(dateString).toLocaleDateString("en-GB", options);  
  }  
}
```

```
render() {
  this.shadowRoot.innerHTML = `
    <h2>${this.driver.givenName} ${this.driver.familyName}</h2>
    <ul>
      <li>Driver code: ${this.driver.code}</li>
      <li>Permanent number: ${this.driver.permanentNumber}</li>
      <li>Nationality: ${this.driver.nationality}</li>
      <li>Date of birth: ${this.readableDate(this.driver.dateOfBirth)}</li>
    </ul>
  `;
}

customElements.define("driver-detail", DriverDetail);
```

```
render() {
  this.shadowRoot.innerHTML = `
    <h2>${this.driver.givenName} ${this.driver.familyName}</h2>
    <ul>
      <li>Driver code: ${this.driver.code}</li>
      <li>Permanent number: ${this.driver.permanentNumber}</li>
      <li>Nationality: ${this.driver.nationality}</li>
      <li>Date of birth: ${this.readableDate(this.driver.dateOfBirth)}</li>
    </ul>
  `;
}

customElements.define("driver-detail", DriverDetail);
```

```
import "./driver-list.js";
import "./driver-detail.js";

const driverListComponent = document.querySelector("driver-list");
const driverDetailComponent = document.querySelector("driver-detail");

driverListComponent.addEventListener("selectDriver", (e) => {
  console.info(e);
  driverDetailComponent.driver = e.detail
}) ;

fetch("http://ergast.com/api/f1/2023/drivers.json")
  .then((res) => res.json())
  .then((data) => {
    const drivers = data.MRData.DriverTable.Drivers;
    driverListComponent.drivers = drivers;
    driverDetailComponent.driver = drivers[0];
  }) ;
```

Component Interaction | Web C X +

← → C 127.0.0.1:3000/15-component-interaction/index.html

Elements Console Sources Network > | ⚙️ ⚙️ X

Filter All levels ▾ No Issues ⚙️

# Component Interaction

Albon, Alexander ▾

Alexander Albon

Driver code: ALB

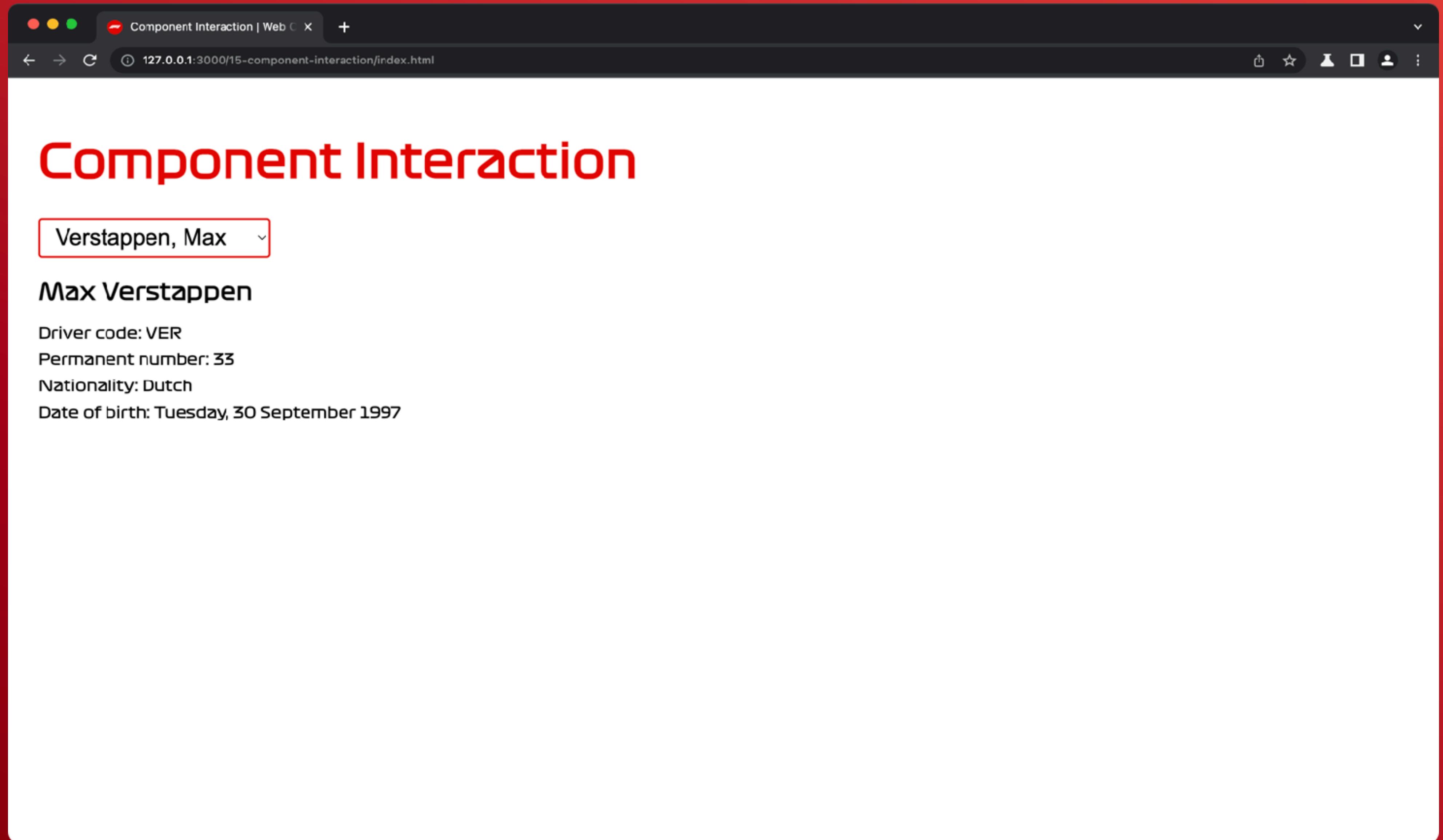
Permanent number: 23

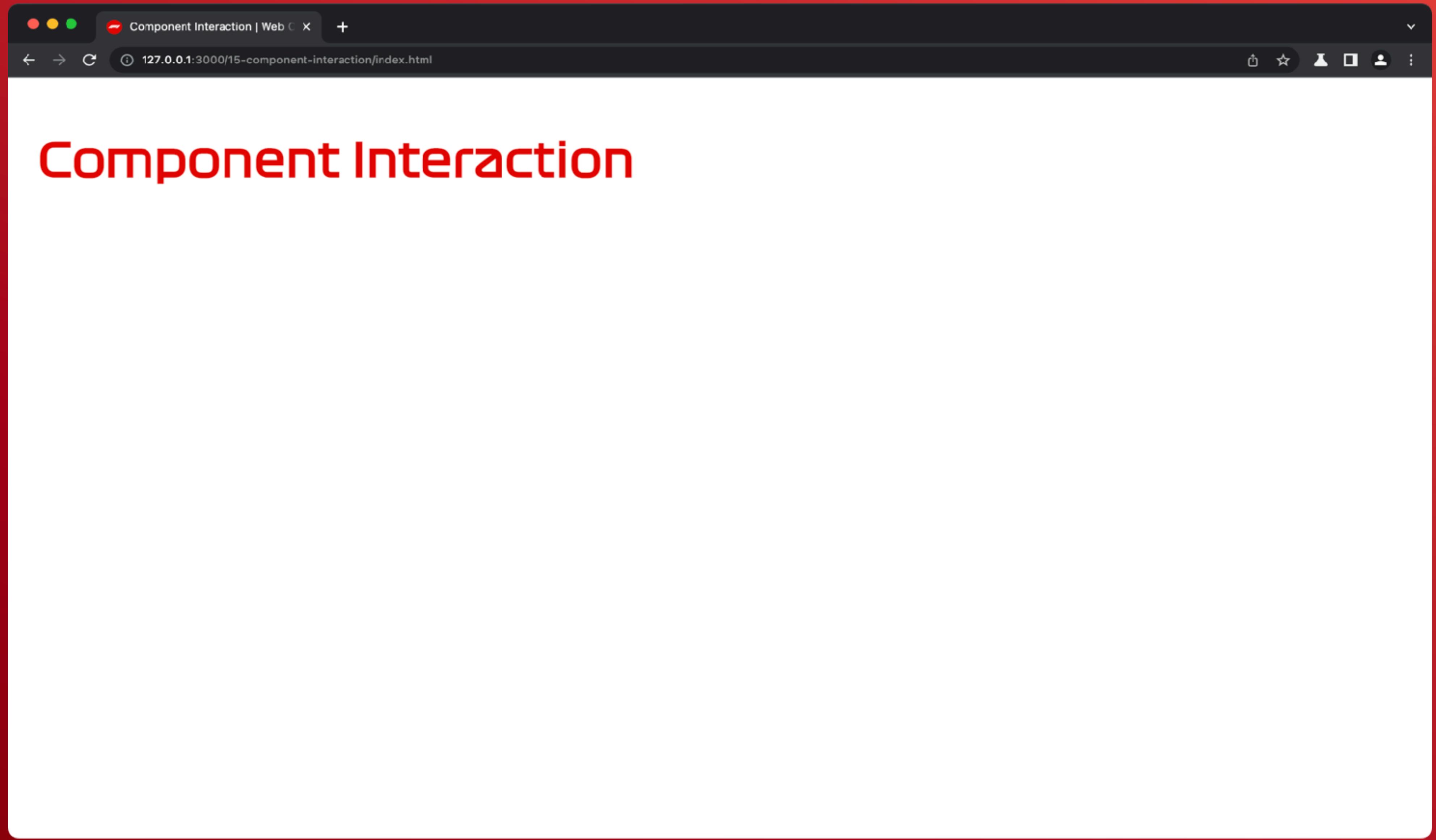
Nationality: Thai

Date of birth: Saturday, 23 March 1996

# JAVASCRIPT

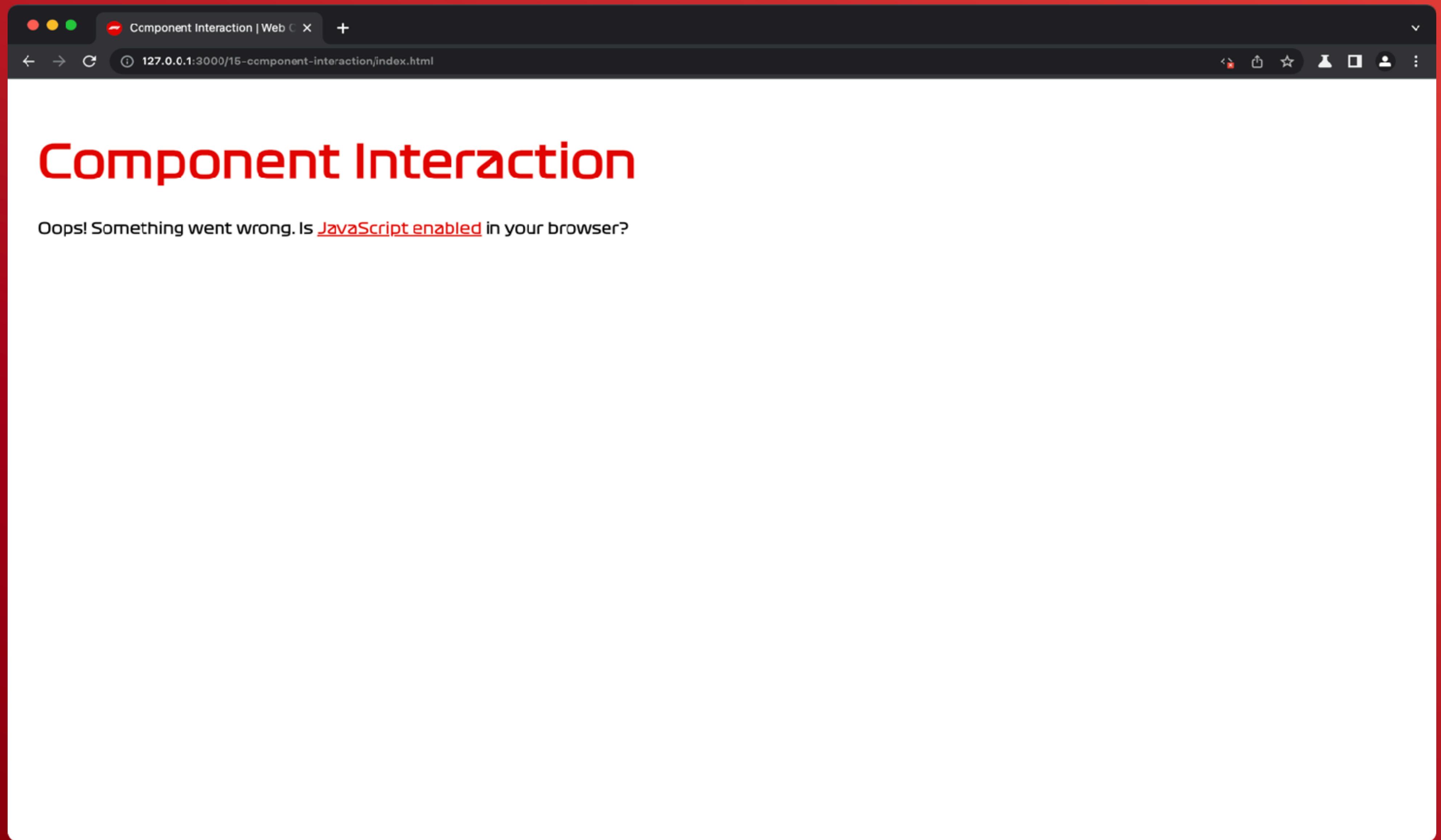






```
<driver-list>Oops! Something went wrong. Is <a href="https://  
www.whatismybrowser.com/guides/how-to-enable-javascript/">JavaScript  
enabled</a> in your browser?</driver-list>
```

```
<noscript>Oops! Something went wrong. Is <a href="https://  
www.whatismybrowser.com/guides/how-to-enable-javascript/">JavaScript  
enabled</a> in your browser?</noscript>
```



# VIDEO COMPONENT

```
<!-- YouTube embed -->
<iframe scrolling="no" marginheight="0" marginwidth="0"
type="text/html" src="https://www.youtube.com/embed/
Q0oeeI3daj4?
autoplay=0&fs=0&iv_load_policy=3&showinfo=0&rel
=0&cc_load_policy=0&start=0&end=0&vq=hd1080&
;origin=https://youtubeembedcode.com" width="640" height="360"
frameborder="0"></ iframe>

<!-- Vimeo embed -->
<iframe allow="autoplay; fullscreen; picture-in-picture"
allowfullscreen="" src="https://player.vimeo.com/video/
705066644" width="640" height="360" frameborder="0"> </ iframe>

<!-- HTML5 video -->
<video controls="" width="640" height="360">
  <source src="max-vs-godzilla.mp4" type="video/mp4" />
</ video>
```

```
<!-- YouTube embed -->
<video-component youtube="Q0oeeI3daj4"></video-component>

<!-- Vimeo embed -->
<video-component vimeo="705066644"></video-component>

<!-- HTML5 video -->
<video-component html5="max-vs-godzilla.mp4"></video-component>

<script src="video-component.js"></script>
```

```
const template = document.createElement("template");
template.innerHTML = `<div></div>`;

class VideoComponent extends HTMLElement {
  static get observedAttributes() {
    return ["youtube", "vimeo", "html5"];
  }

  set youtube(value) {
    this.videoElement.innerHTML = `
      <iframe frameborder="0" scrolling="no" marginheight="0"
marginwidth="0" width="640" height="360" type="text/html"
      src="https://www.youtube.com/embed/${value}?
autoplay=0&fs=0&iv_load_policy=3&showinfo=0&rel=0&cc_load_polic
y=0&start=0&end=0&vq=hd1080&origin=https://
youtubeembedcode.com">
      </iframe>
    `;
  }
}
```

```
const template = document.createElement("template");
template.innerHTML = `<div></div>`;

class VideoComponent extends HTMLElement {
  static get observedAttributes() {
    return ["youtube", "vimeo", "html5"];
  }

  set youtube(value) {
    this.videoElement.innerHTML = `
      <iframe frameborder="0" scrolling="no" marginheight="0"
marginwidth="0" width="640" height="360" type="text/html"
      src="https://www.youtube.com/embed/${value}?
autoplay=0&fs=0&iv_load_policy=3&showinfo=0&rel=0&cc_load_polic
y=0&start=0&end=0&vq=hd1080&origin=https://
youtubeembedcode.com">
      </iframe>
    `;
  }
}
```

```
set vimeo(value) {
  this.videoElement.innerHTML = `
    <iframe width="640" height="360" frameborder="0"
allow="autoplay; fullscreen; picture-in-picture"
allowfullscreen src="https://player.vimeo.com/video/${value}">
    </iframe>
  `;
}

set html5(value) {
  this.videoElement.innerHTML = `
    <video controls width="640" height="360">
      <source src="${value}" type="video/mp4" />
    </video>
  `;
}
```

```
set vimeo(value) {
  this.videoElement.innerHTML = `
    <iframe width="640" height="360" frameborder="0"
allow="autoplay; fullscreen; picture-in-picture"
allowfullscreen src="https://player.vimeo.com/video/${value}">
    </iframe>
  `;
}

set html5(value) {
  this.videoElement.innerHTML = `
    <video controls width="640" height="360">
      <source src="${value}" type="video/mp4" />
    </video>
  `;
}
```

```
constructor() {  
    super();  
    this.attachShadow({ mode: "open" });  
    this.shadowRoot.appendChild(template.content.cloneNode(true));  
    this.videoElement = this.shadowRoot.querySelector("div");  
}  
  
attributeChangedCallback(attrName, oldValue, newValue) {  
    switch (attrName) {  
        case "youtube":  
            this.youtube = newValue;  
            break;  
        case "vimeo":  
            this.vimeo = newValue;  
            break;  
        case "html5":  
            this.html5 = newValue;  
            break;  
    }  
}  
}  
  
customElements.define("video-component", VideoComponent);
```

```
constructor() {  
  super();  
  this.attachShadow({ mode: "open" });  
  this.shadowRoot.appendChild(template.content.cloneNode(true));  
  this.videoElement = this.shadowRoot.querySelector("div");  
}  
  
attributeChangedCallback(attrName, oldValue, newValue) {  
  switch (attrName) {  
    case "youtube":  
      this.youtube = newValue;  
      break;  
    case "vimeo":  
      this.vimeo = newValue;  
      break;  
    case "html5":  
      this.html5 = newValue;  
      break;  
  }  
}  
}  
  
customElements.define("video-component", VideoComponent);
```

index.html – code

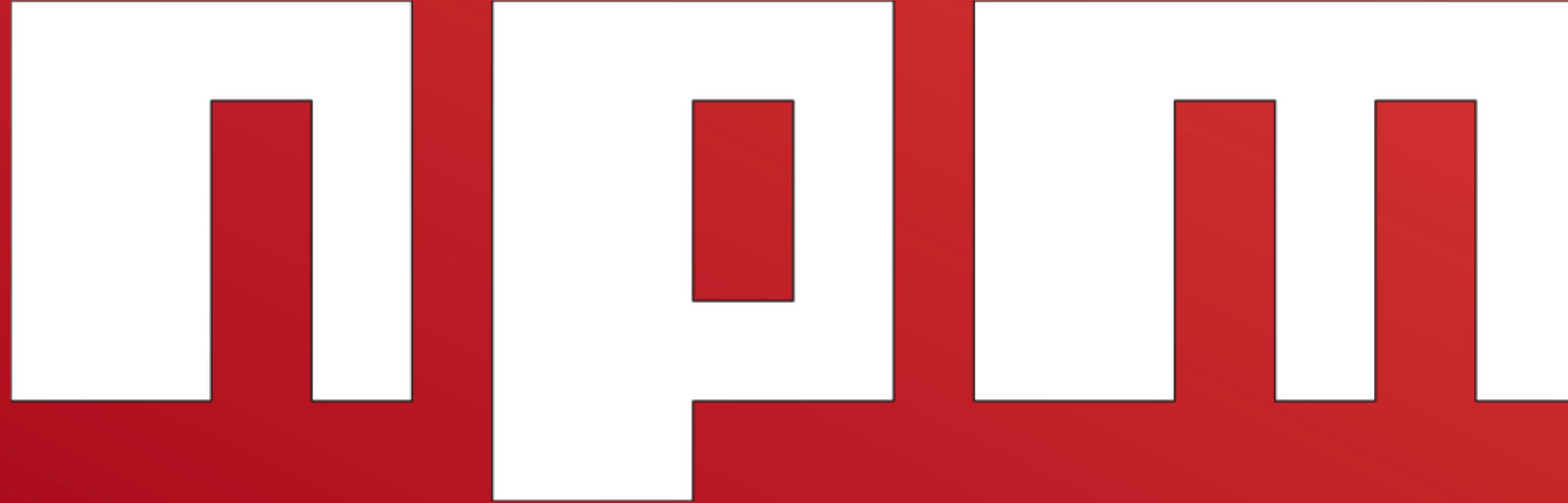
index.html X

16-video-component > index.html > ...

```
1  <!DOCTYPE html>
2  <html lang="en">
3    <head>
4      <meta charset="UTF-8" />
5      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
6      <title>Video Component | Web Components</title>
7      <link rel="stylesheet" href="/assets/css/style.css" />
8    </head>
9    <body>
10   <h1>Video Component</h1>
11   <sub>run this demo in a browser, not in VS Code live preview</sub>
12   <!-- YouTube embed -->
13   <video-component youtube="J7Xpy9AuC-A"></video-component>
14   <!-- Vimeo embed -->
15   <video-component vimeo="817910772?h=262663fd68"></video-component>
16   <!-- HTML5 video -->
17   <video-component html5="toto-wolff.mp4"></video-component>
18   <script src="video-component.js"></script>
19 </body>
20 </html>
21
```

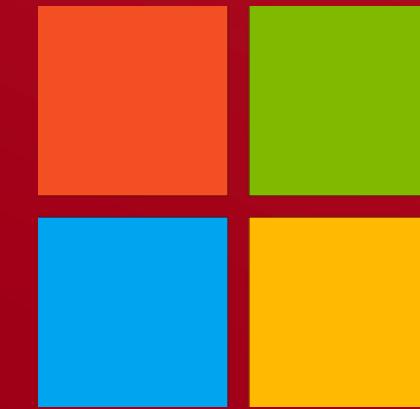
feature/refactor ⌂ ⌂ 0 △ 0

Ln 1, Col 1 Spaces: 2 UTF-8 LF HTML ⚙ Port: 3000 ✓ Spell ✎ Prettier



# Node Package Manager

The free npm registry is the center of JavaScript code sharing, and with more than two million packages, the largest software registry in the world.



```
<!-- YouTube embed -->
<video-component youtube="Q0oeeI3daj4"></video-component>

<!-- Vimeo embed -->
<video-component vimeo="705066644"></video-component>

<!-- HTML5 video -->
<video-component html5="max-vs-godzilla.mp4"></video-component>

<script src="video-component.js"></script>
```

```
<!-- YouTube embed -->
<video-component youtube="Q0oeeI3daj4"></video-component>

<!-- Vimeo embed -->
<video-component vimeo="705066644"></video-component>

<!-- HTML5 video -->
<video-component html5="max-vs-godzilla.mp4"></video-component>

<script src="https://unpkg.com/@rowdy.rabouw/video-component/src/video-component.js"></script>
```

unpkg is a fast, global CDN (Content Delivery Network) for everything on npm



```
"dependencies": {  
    "@rowdy.rabouw/video-component": "^1.0.0",  
    "core-js": "^3.8.3",  
    "vue": "^3.2.13"  
},
```

```
import { createApp } from "vue";
import App from "./App.vue";
import "@rowdy.rabouw/video-component";

createApp(App).mount("#app");
```

```
<template>
  
  <video-component youtube="J7Xpy9AuC-A"></video-component>
  <HelloWorld msg="Welcome to Your Vue.js App" />
</template>
```

vue-video-app

localhost:8080

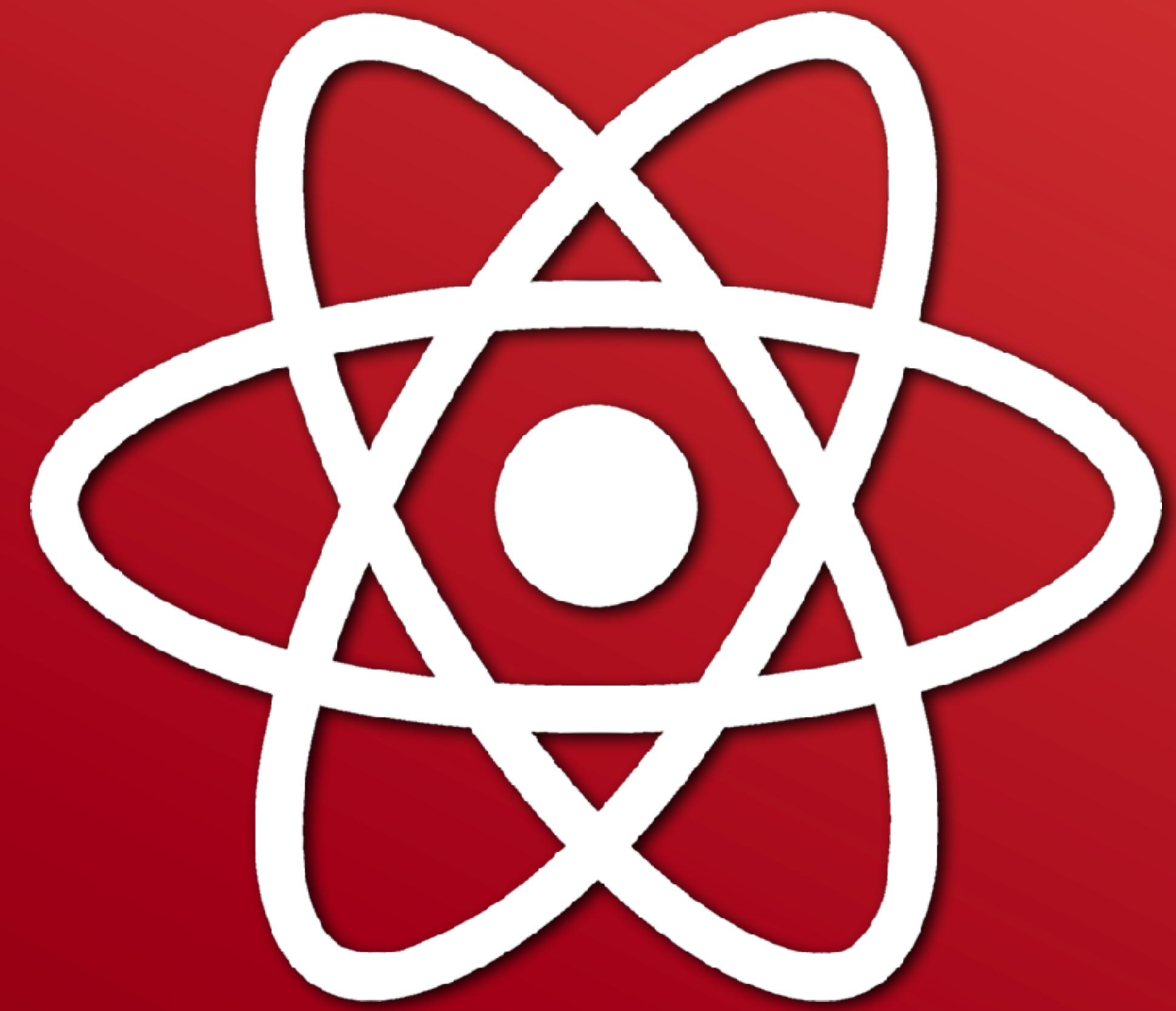


Welcome to Your Vue.js App

For a guide and recipes on how to configure / customize this project,  
check out the [vue-cli documentation](#).

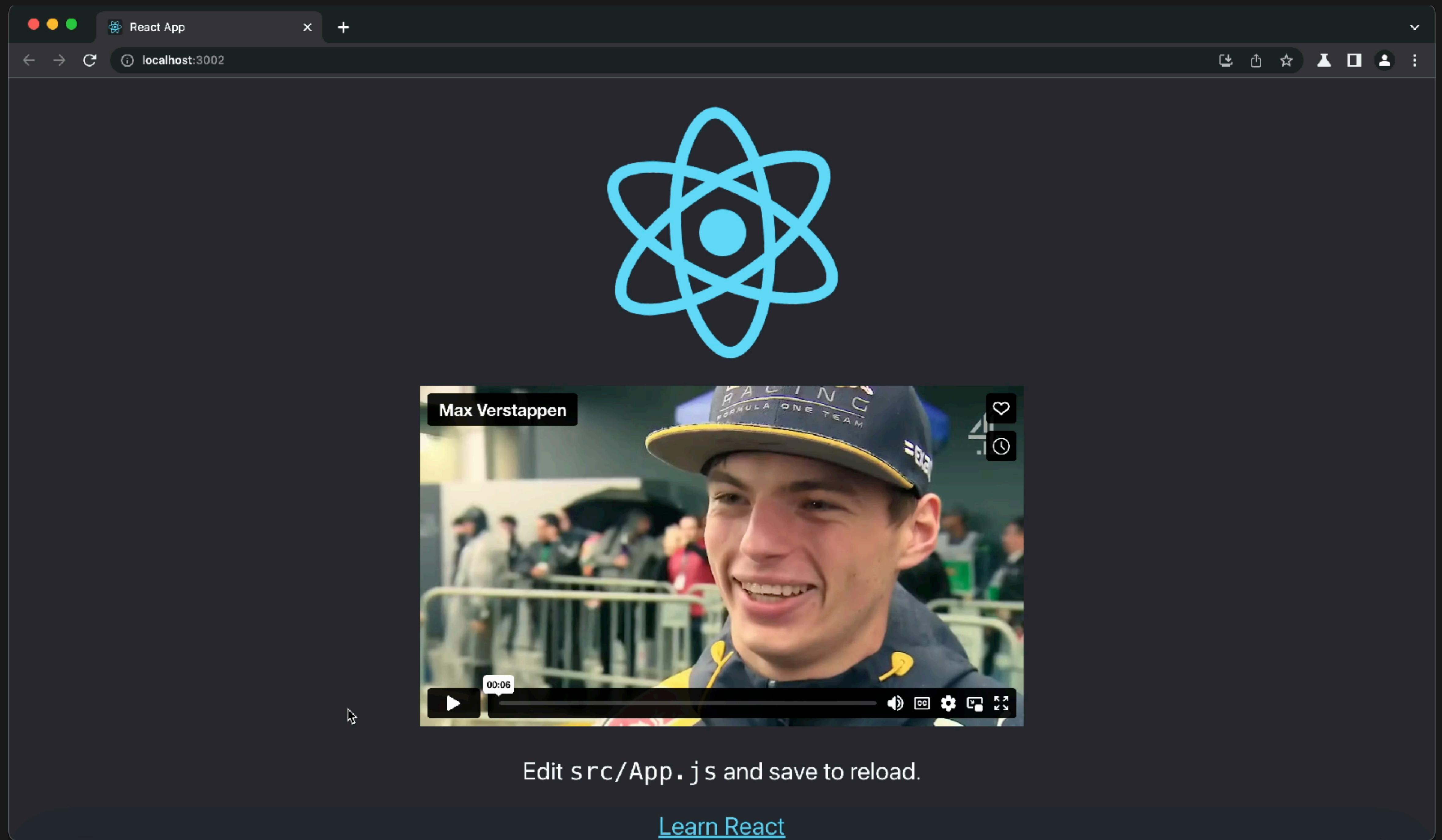
Installed CLI Plugins

babel eslint



```
"dependencies": {  
    "@rowdy.rabouw/video-component": "^1.0.0",  
    "@testing-library/jest-dom": "^5.16.4",  
    "@testing-library/react": "^13.1.1",  
    "@testing-library/user-event": "^13.5.0",  
    "react": "^18.1.0",  
    "react-dom": "^18.1.0",  
    "react-scripts": "5.0.1",  
    "web-vitals": "^2.1.4"  
},
```

```
function App() {
  return (
    <div className="App">
      <header className="App-header">
        <img src={logo} className="App-logo" alt="logo" />
        <video-component vimeo="817910772?h=262663fd68"></video-component>
        <p>
          Edit <code>src/App.js</code> and save to reload.
        </p>
        <a
          className="App-link"
          href="https://reactjs.org"
          target="_blank"
          rel="noopener noreferrer"
        >
          Learn React
        </a>
      </header>
    </div>
  );
}
```





```
"dependencies": {  
    "@angular/animations": "~13.0.0",  
    "@angular/common": "~13.0.0",  
    "@angular/compiler": "~13.0.0",  
    "@angular/core": "~13.0.0",  
    "@angular/forms": "~13.0.0",  
    "@angular/platform-browser": "~13.0.0",  
    "@angular/platform-browser-dynamic": "~13.0.0",  
    "@angular/router": "~13.0.0",  
    "@rowdy.rabouw/video-component": "^1.0.0",  
    "rxjs": "~7.4.0",  
    "tslib": "^2.3.0",  
    "zone.js": "~0.11.4"  
},
```

```
import { NgModule, CUSTOM_ELEMENTS_SCHEMA } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import '@rowdy.rabouw/video-component';

import { AppComponent } from './app.component';

@NgModule({
  declarations: [AppComponent],
  imports: [BrowserModule],
  providers: [],
  bootstrap: [AppComponent],
  schemas: [CUSTOM_ELEMENTS_SCHEMA],
})
export class AppModule {}
```

```
<svg  
    id="rocket-smoke"  
    xmlns="http://www.w3.org/2000/svg"  
    width="516.119"  
    height="1083.632"  
    viewBox="0 0 516.119 1083.632"  
>  
  <title>Rocket Ship Smoke</title>  
  <path  
      id="Path_40"  
      data-name="Path 40"  
      d="..."  
      fill="#f5f5f5"  
  />  
  </svg>  
</div>  
  
<video-component html5="assets/toto-wolff.mp4"></video-component>  
  
<!-- Resources --&gt;<br/><h2>Resources</h2>  
<p>Here are some links to help you get started:</p>
```

add toto-wolff.mp4 to src/assets

A AngularVideoApp

localhost:4200

Welcome

angular-video-app app is running!

0:00 / 0:03

## Resources

Here are some links to help you get started:

- [Learn Angular >](#)
- [CLI Documentation >](#)
- [Angular Material >](#)
- [Angular Blog >](#)
- [Angular DevTools >](#)

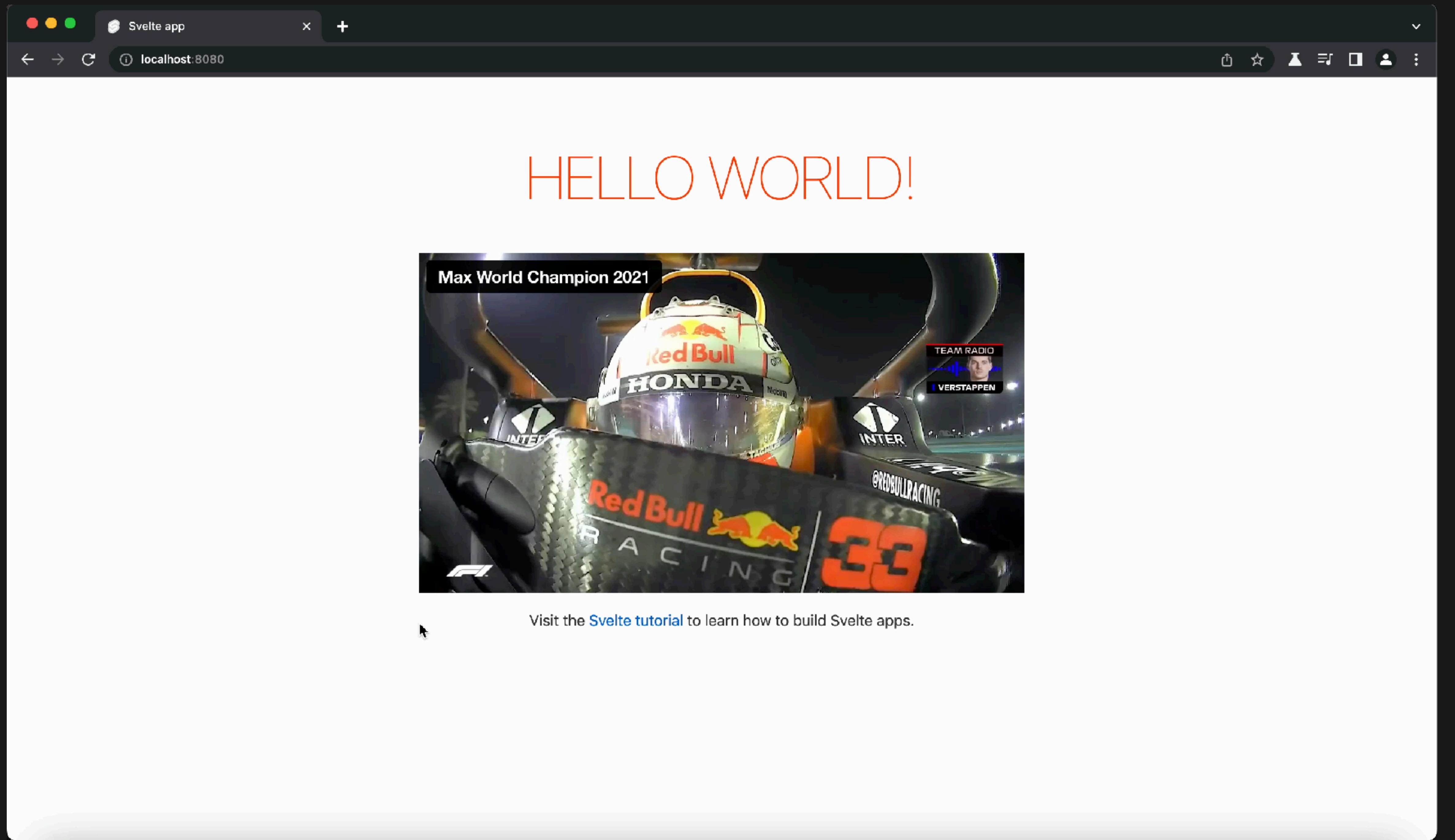
## Next Steps



```
"dependencies": {  
    "@rowdy.rabouw/video-component": "^1.0.0",  
    "sirv-cli": "^2.0.0"  
},
```

```
<script>
  import '@rowdy.rabouw/video-component';
  export let name;
</script>

<main>
  <h1>Hello {name}!</h1>
  <video-component vimeo="817915307?h=3c213cc142"></video-component>
  <p>Visit the <a href="https://svelte.dev/tutorial">Svelte tutorial</a>
to learn how to build Svelte apps.</p>
</main>
```





# DĚKUJI MNOHOKRÁT

<https://rowdy.codes/webexpo>