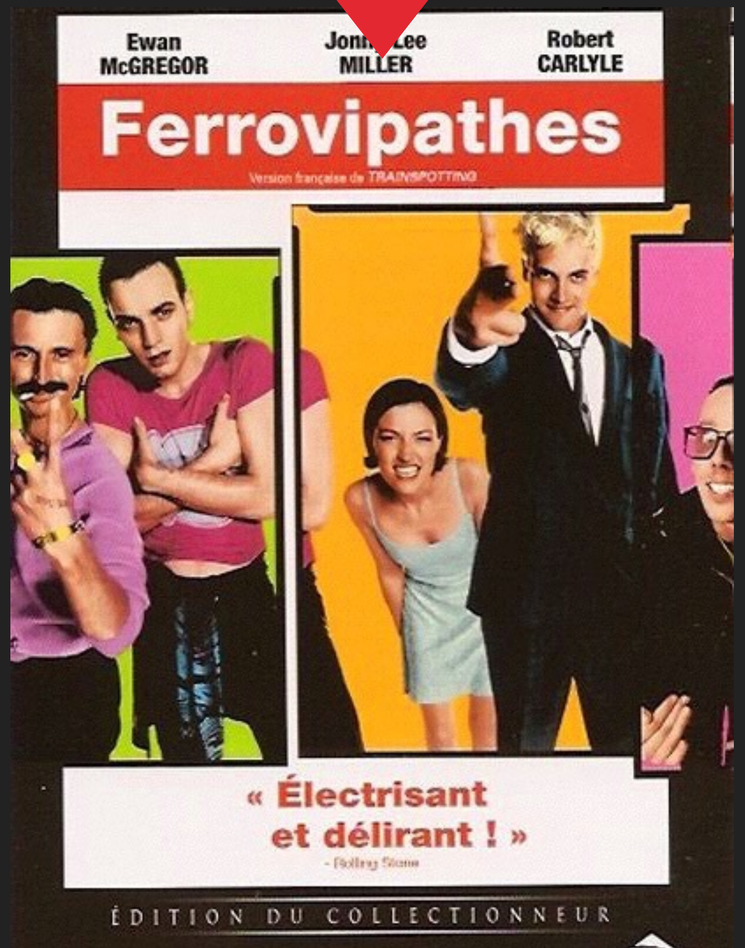
TOUTE LA LUMIÈRE SUR LE

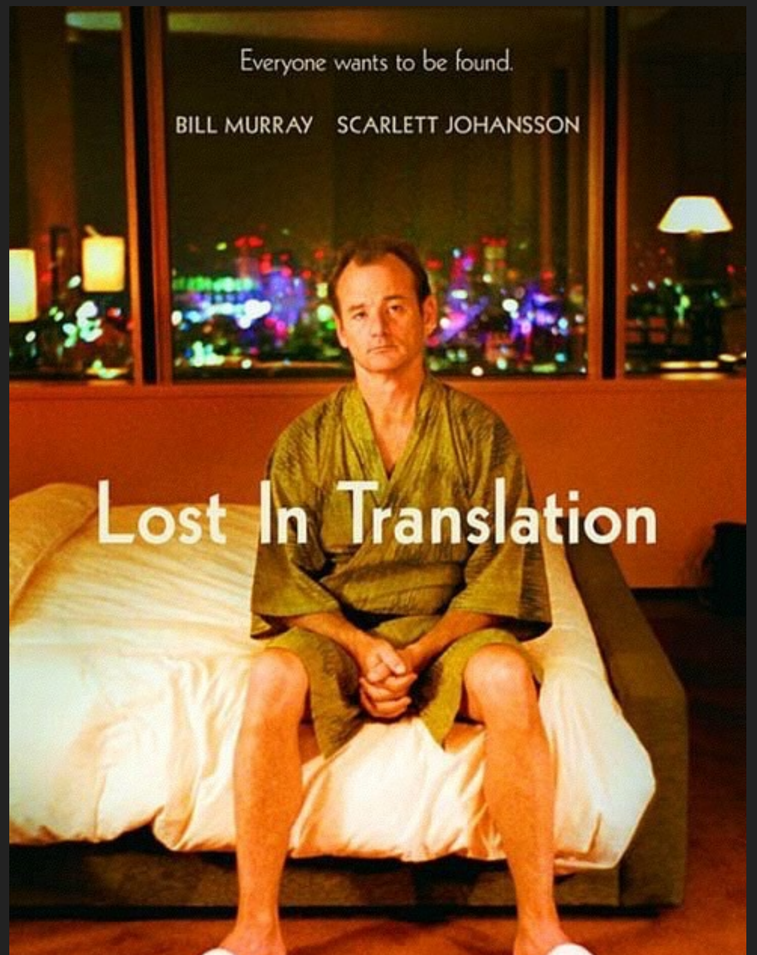
SHADOW DOM

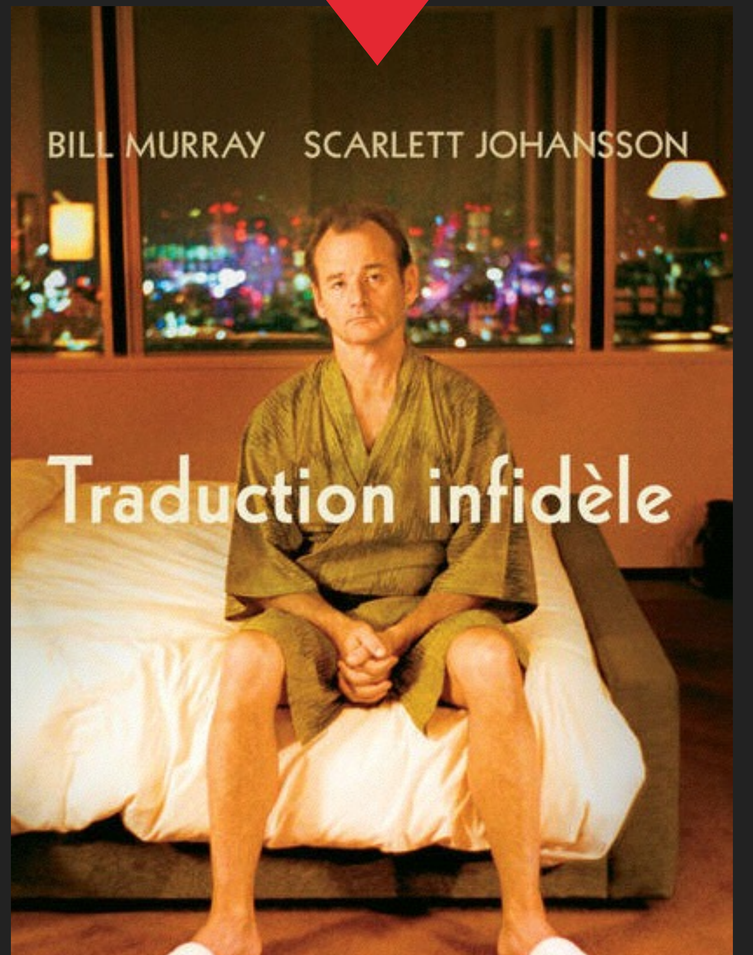
CONFOO MONTRÉAL, RODOLPHE RIMELÉ, MARS 2017 @DIOU

POUR MOI, LE QUÉBEC C'EST AVANT TOUT DU COURAGE DANS LE LANGAGE...



POUR MOI, LE QUÉBEC C'EST AVANT TOUT DU COURAGE DANS LE LANGAGE...



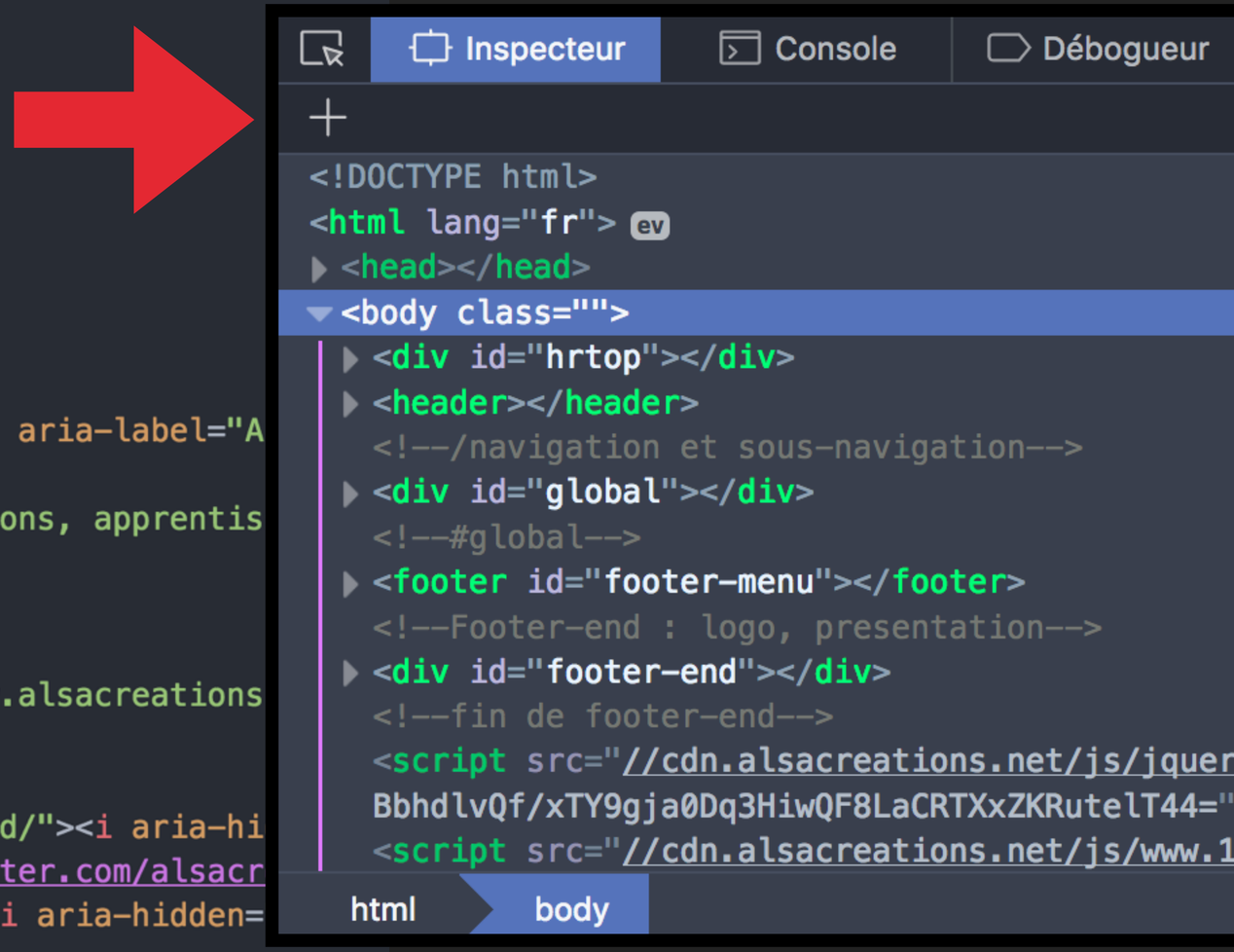


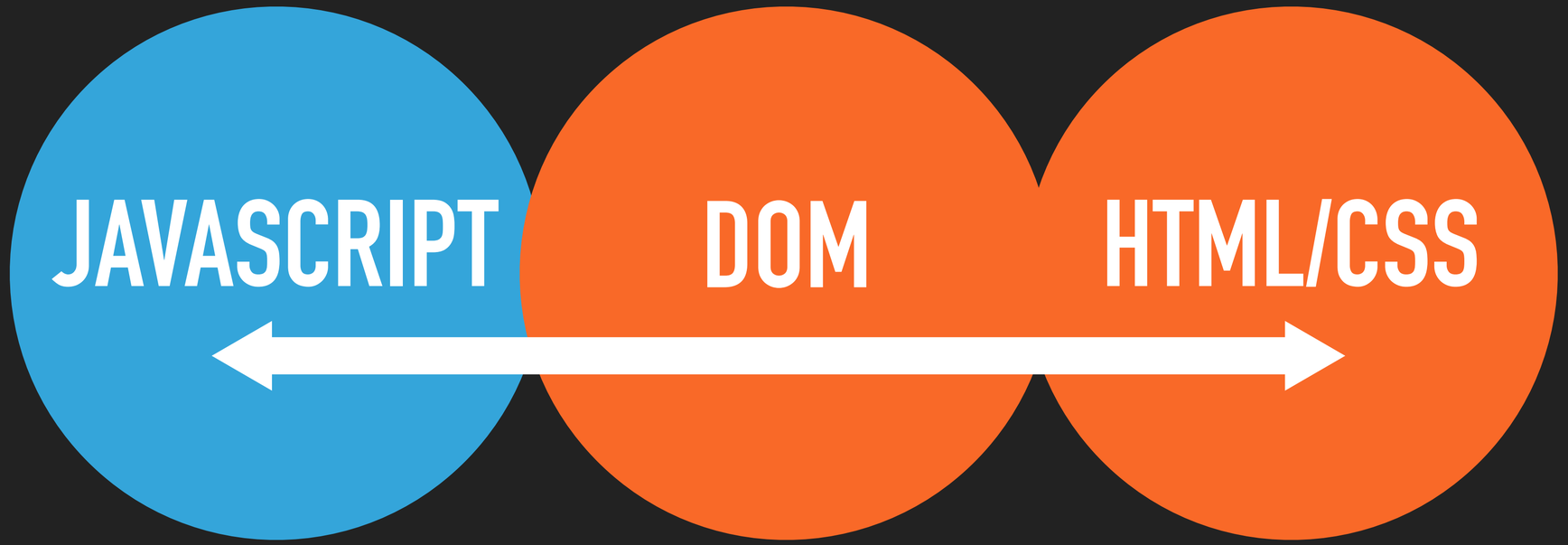
DOCUMENT OBJECT MODEL

LE DOM

HTML DOM

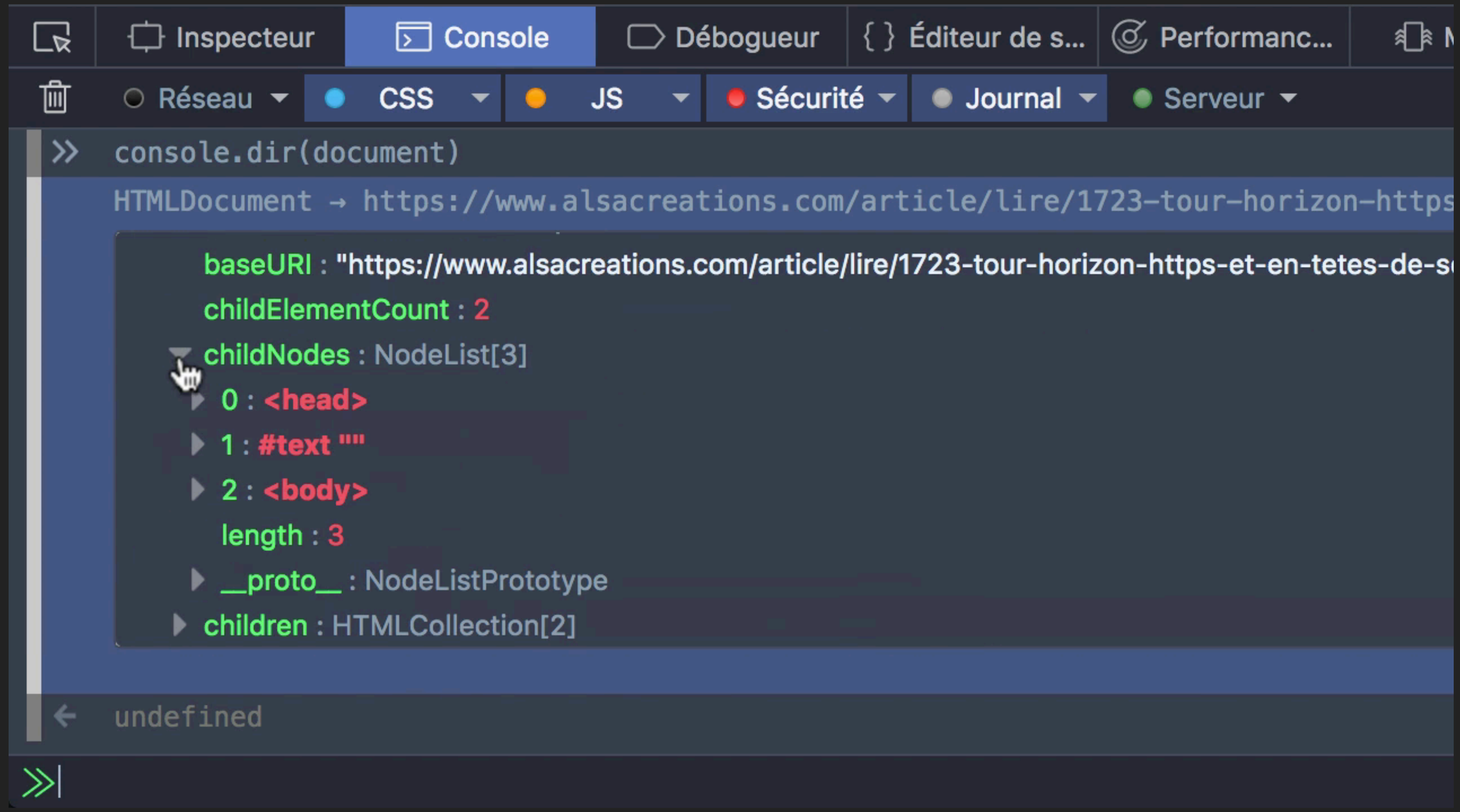
```
<!DOCTYPE html>

<html lang="fr">
<head></head>
<body class="">

<div id="hrtop"><div class="arf"></div></div>

<header>
<div id="header">

  <ul id="evitement">
    <li><a href="#menu">Aller au menu</a></li>
    <li><a href="#global">Aller au contenu</a></li>
    <li><a href="#search-q">Aller à la recherche</a></li>
  </ul> <!-- /evitement -->

  <div class="inside" role="banner">
    <button class="burger js-burger js-hidden" type="button" aria-label="A

    <h1><a href="//www.alsacreations.com/" title="Alsacréations, apprentis

    <nav role="navigation" class="megamenu">
      <ul id="menu" class="nav-menu">
        <li class="nav-item menu-home active"><a href="//www.alsacreations
        <div class="sub-nav">
          <ul class="sub-nav-group">
            <li><a href="//www.alsacreations.com/actu/add/"><i aria-hi
            <li><a rel="noopener" href="https://www.twitter.com/alsacr
            <li><a href="//www.alsacreations.com/rss/"><i aria-hidden=
          </ul>
        </div>
```





spécifiés par… ECMA W3C

DOM TREE : ARBRE D'OBJETS EN MÉMOIRE

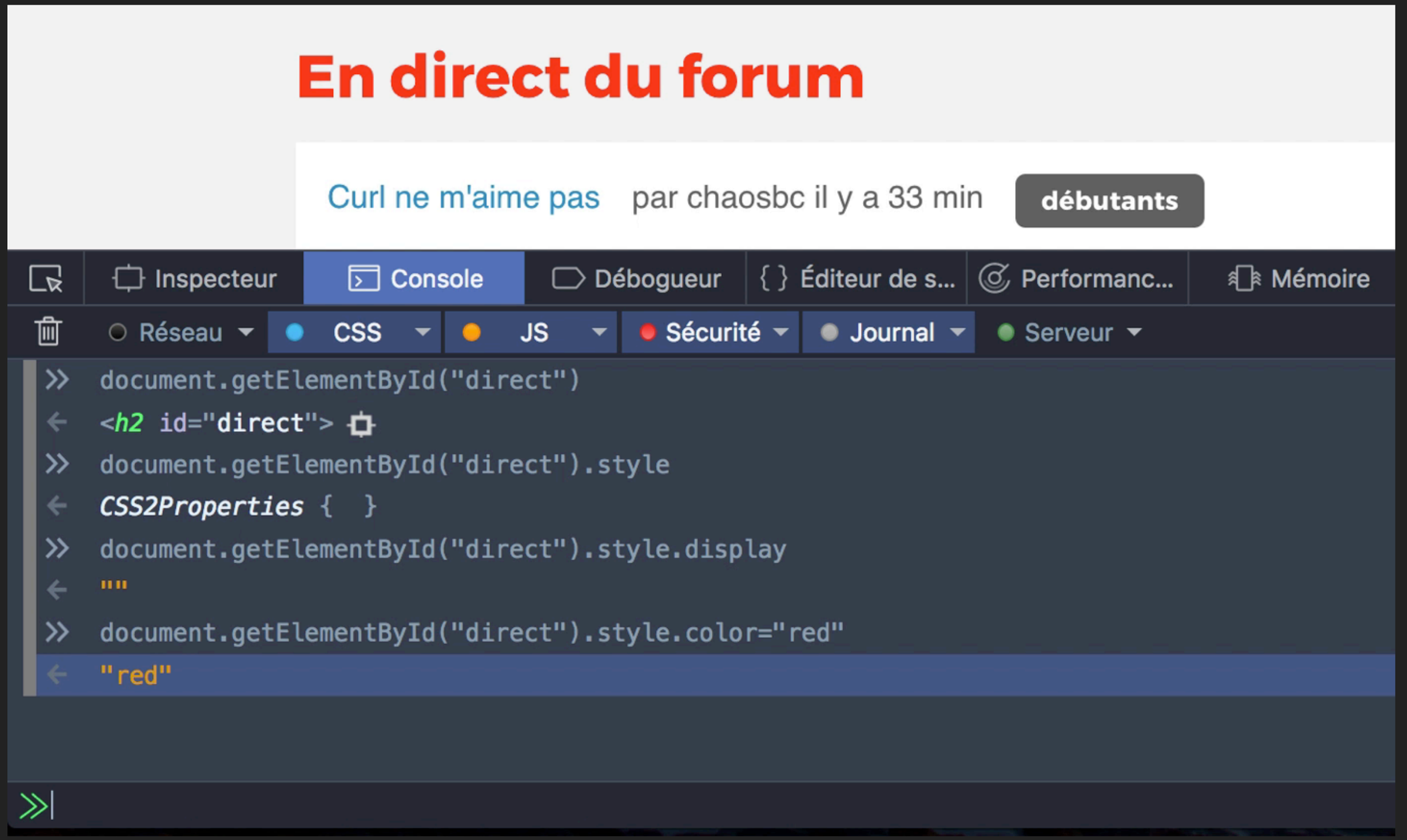


C'EST AUSSI CE QUI EST EXPLOITÉ POUR L'ACCESSIBILITÉ

Par exemple :

- utiliser la sémantique HTML des éléments et ne pas les détourner de leur rôle initial
  (un lien ou un bouton sont « cliquables », une image non)
- donner un alt aux images de contenu
- si nécessaire utiliser ARIA (WAI)

DOM TREE : INTERACTION AVEC JAVASCRIPT



VOUS LE VOYEZ LÀ ?

LE SHADOW DOM

C'EST UNE SPEC W3C

https://www.w3.org/TR/shadow-dom/

...QUI FAIT PARTIE DES

WEB COMPONENTS

- SHADOW DOM = ENCAPSULATION
- TEMPLATES = BOUTS DE DOM
- CUSTOM ELEMENTS = EXTENSIONS
- HTML IMPORTS = « PACKAGING »

SUPPORT NAVIGATEUR

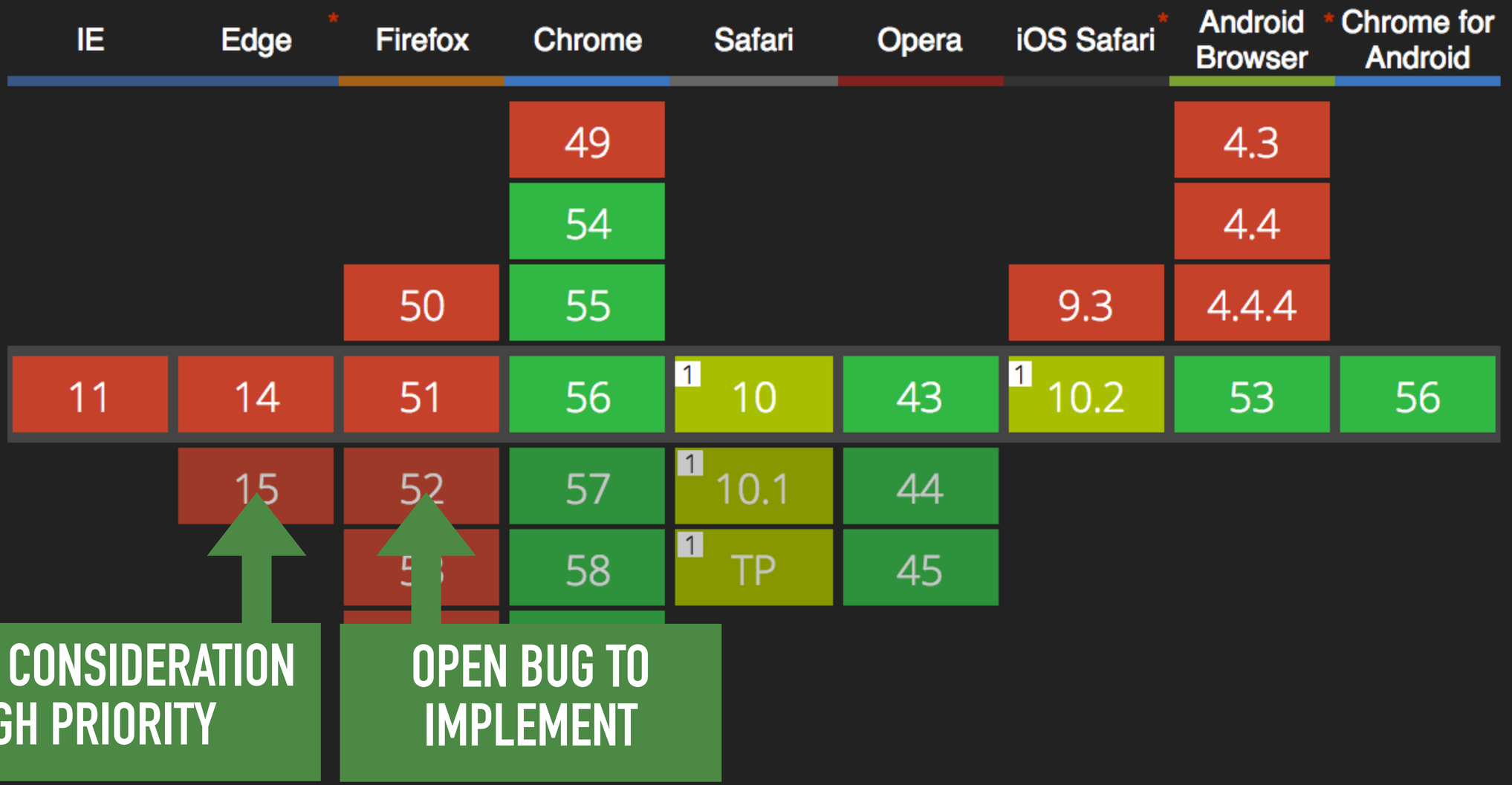
DEUX VERSIONS
V0 ET V1

QUELQUES CAS SPÉCIAUX :
SAMSUNG BROWSER, ANDROID 4.4.X

D'ACTUALITÉ
EN GROS, MÊME PRINCIPE

Shadow DOM v1

Global 49.04% + 9.89% = 58.94%



Browser support	CHROME	OPERA	FIREFOX	SAFARI	EDGE
TEMPLATES	STABLE	STABLE	STABLE	STABLE	STABLE
IMPORTS	STABLE	STABLE	POLYFILL ON HOLD	POLYFILL ON HOLD	POLYFILL CONSIDERING
CUSTOM ELEMENTS	STABLE	STABLE	POLYFILL DEVELOPING	POLYFILL DEVELOPING	POLYFILL CONSIDERING
SHADOW DOM	STABLE	STABLE	POLYFILL DEVELOPING	STABLE	POLYFILL CONSIDERING

SHIMS*

ShadyDOM
https://github.com/webcomponents/shadydom
et ShadyCSS...

* petits bouts de code Javascript pour combler le manque de support natif

```
bower install --save webcomponents/shadydom
bower install --save webcomponents/shadycss
```

Web Components

https://www.webcomponents.org

www.webcomponents.org

Repositories | People 18

Pinned repositories

webcomponentsjs

A suite of polyfills supporting the HTML Web Components specs

JavaScript ★ 2k 307

custom-elements

A polyfill for HTML Custom Elements v1

JavaScript ★ 100 25

shadydom

ShadowDOM v1 shim

HTML ★ 42 15

react-integration

Converts web components into React components so that you can use them as first class citizens in your React components.

JavaScript ★ 124 16

webcomponents.org

HTML ★ 59 27

#UseThePlatform to build a better web

The Polymer Project helps you deliver amazing user experiences by unlocking the full potential of the web platform.

Use the Polymer library to make the most of Web Components, a powerful new platform feature for extending HTML and componentizing your apps.

Use the Polymer App Toolbox to build cutting-edge Progressive Web Apps that load quickly, respond instantly and work from anywhere.

GET STARTED | TRY THE SHOP DEMO | ABOUT THE PROJECT

https://www.polymer-project.org

Custom Elements

Explore the world of Web Components

Search...

OR BROWSE ALL REPOSITORIES

https://www.customelements.io/

À QUOI CELA PEUT-IL BIEN SERVIR ?

CAS UTILES

HTML Futures

Joel Nava, Bruce Hunt, T.V. Raman

Adobe Systems Inc.

bstract

dobe believes that the Future of HTML lies in retaining the bulk of the behaviora
gset and associated CSS standard while moving to a fully XML compliant syntax
ke a phased approach as the evolving future of HTML. In the primary phase, HT
accomodate XML. In the succeeding phase, HTML is altered to have a well form
e current syntax(es). In the succeeding phase, the semantics of HTML are expose
rough an XML document object model via CSS so that any tag can access the se
this way, XML integrates with HTML. In the last phase, HTML is subsumed by
ayout behaviors) are retained as a pres tation capability. This approach we belie
igration path for HTML's future.

ntroduction

ll languages grow and change. HTML continues to be a useful content expressio
rowth. It is now time for HTML to mature and grow to support both content drive
the sensate spaces as well. In an ideal world HTML has a simpler more coheran
ontent flow semantics. In that same world, non conflicting custom tag sets would
pplication structure and permit it to be presented in a variety of formats constrain
ontent in a manner best perceived by the content consumer. This would include th
recision rendering in the sensate spaces (visual, auditory, or tactile,or smell!?) dri
yout driven forms. A dynamically drivable precision graphic expression capabili
entify and expose graphics structure along with document structure provides the
yout, but also provides the key to render the graphics in other sensate spaces as v
gh utility growth and permit *standardized* interpreted semantic tag sets, HTML s
ith the work that opens up these new vistas, that is, XML. It is time to plan an en
chieve these goals.

CRÉER SES PROPRES ÉLÉMENTS

(#OLD ON LE PRÉDISAIT DÉJÀ POUR XML EN 1998)

EXEMPLES :

<BI-DULE></BI-DULE>
<MARK-DOWN></MARK-DOWN>

ILS DOIVENT AVOIR DES –

CRÉER DU CODE RÉUTILISABLE

DU CODE COMPLEXE DANS UN ÉLÉMENT « EN APPARENCE » SIMPLE, GÉRER DES ÉTATS...

ENCAPSULER LES STYLES ET LES SCRIPTS

== SCOPE

2.1 BOUTONS

2-BUTTONS.SCSS

Les boutons sont utilisés pour inciter à l'action (CTA). Ils sont majoritairement présents sur les headers des pages pour inviter à acheter, réserver ou se renseigner. Pour des actions moins importantes, l'utilisation d'un lien est privilégiée.

Ils peuvent être utilisés avec toutes les couleurs toniques de la charte.

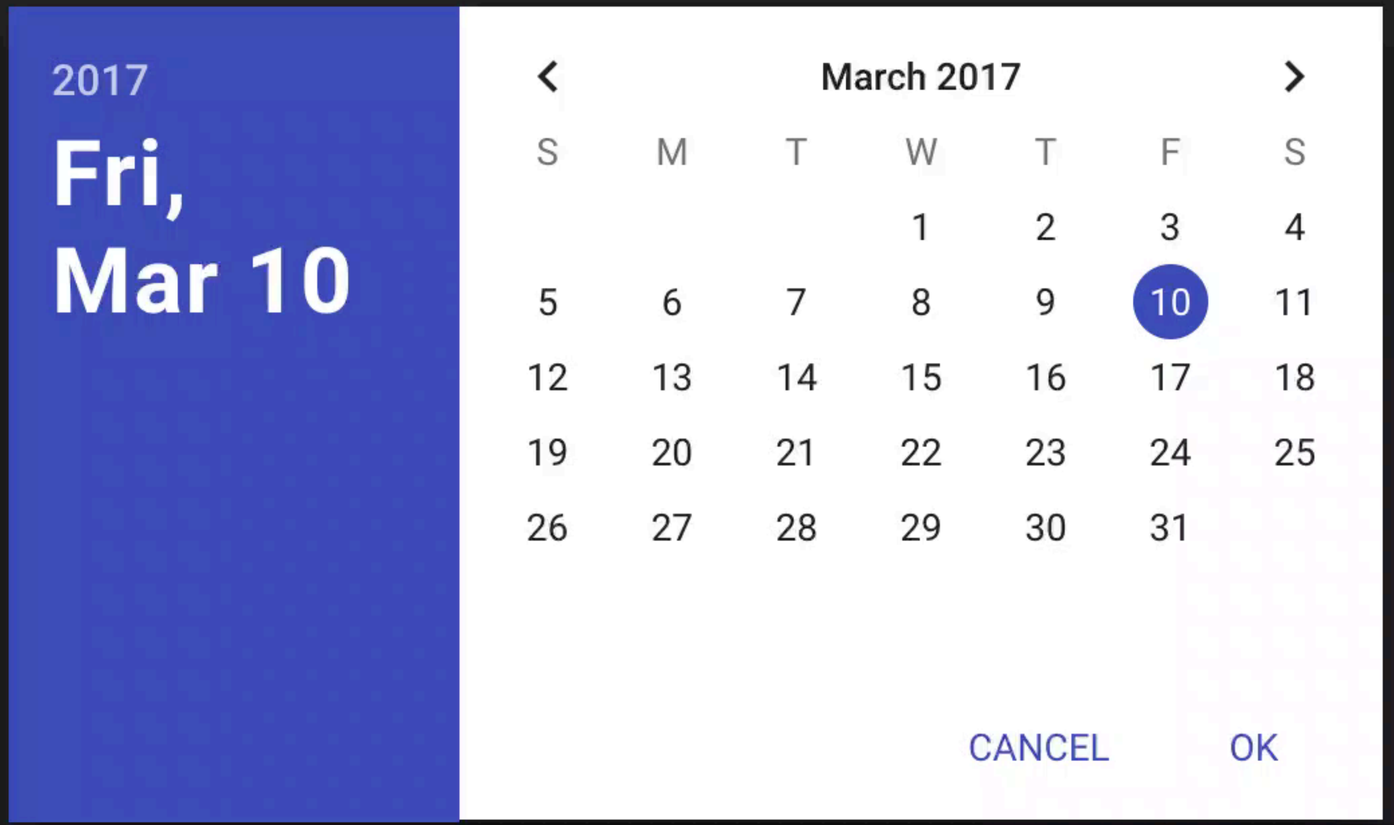
DÉCOUVREZ LES COULEURS DE LA CHARTE ›

.btn-blue - bleu primaire
.btn-teal - bleu canard
.btn-applegreen - vert pomme
.btn-plum - prune
.btn-raspberry - framboise
.btn-purple - violet

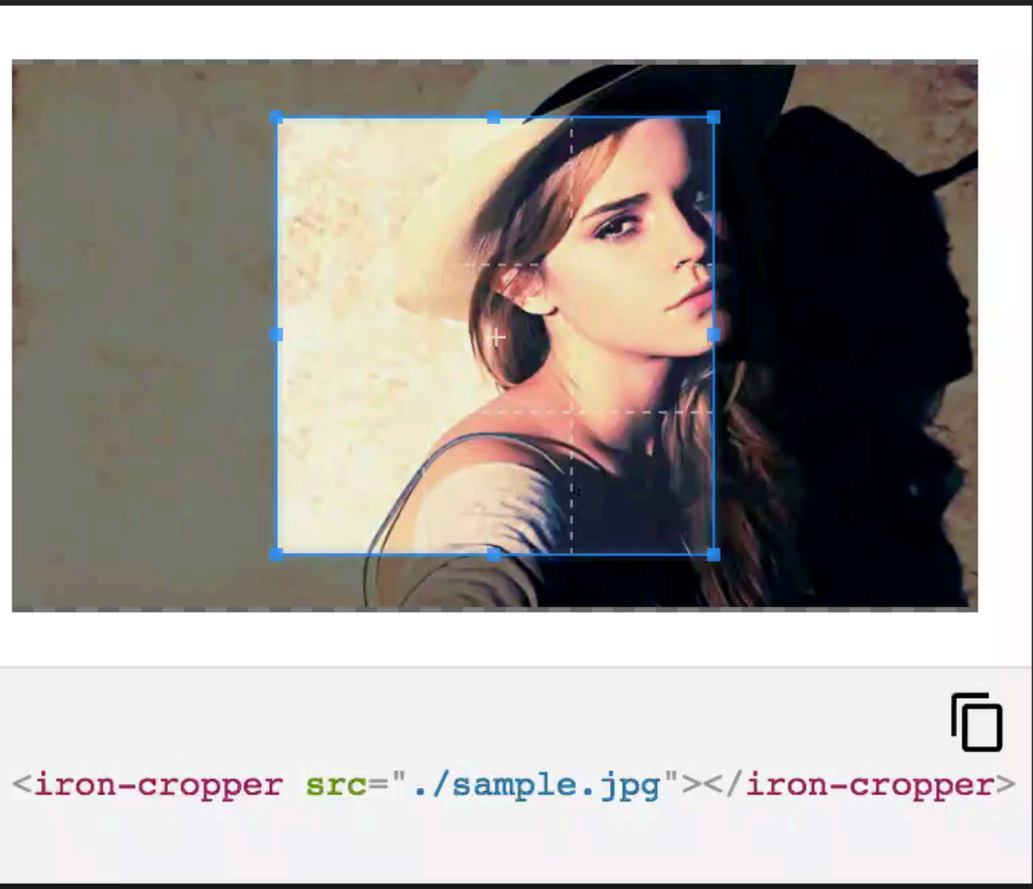
STYLEGUIDE

EXEMPLES DE COMPOSANTS PRATIQUES

WEBCOMPONENTS, POLYMER, CUSTOMELEMENTS…


2017
Fri,
Mar 10
March 2017
S M T W T F S
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31
CANCEL
OK

CE CODE SEMBLE
SI SIMPLE...



```
<iron-cropper src="./sample.jpg"></iron-cropper>
```

Expand Export

```
<vaadin-split-layout>
  <vaadin-split-layout vertical>
    <div>First layout content</div>
    <div>Second layout content</div>
  </vaadin-split-layout>
  <vaadin-split-layout vertical>
    <div>Third layout content</div>
    <div>Fourth layout content</div>
  </vaadin-split-layout>
</vaadin-split-layout>
```

```
<paper-timezone timezone="[[timezone]]" label="Timezone (Continent/City)" error-message="Please enter a timezone offset
```

Dialog can be styled using

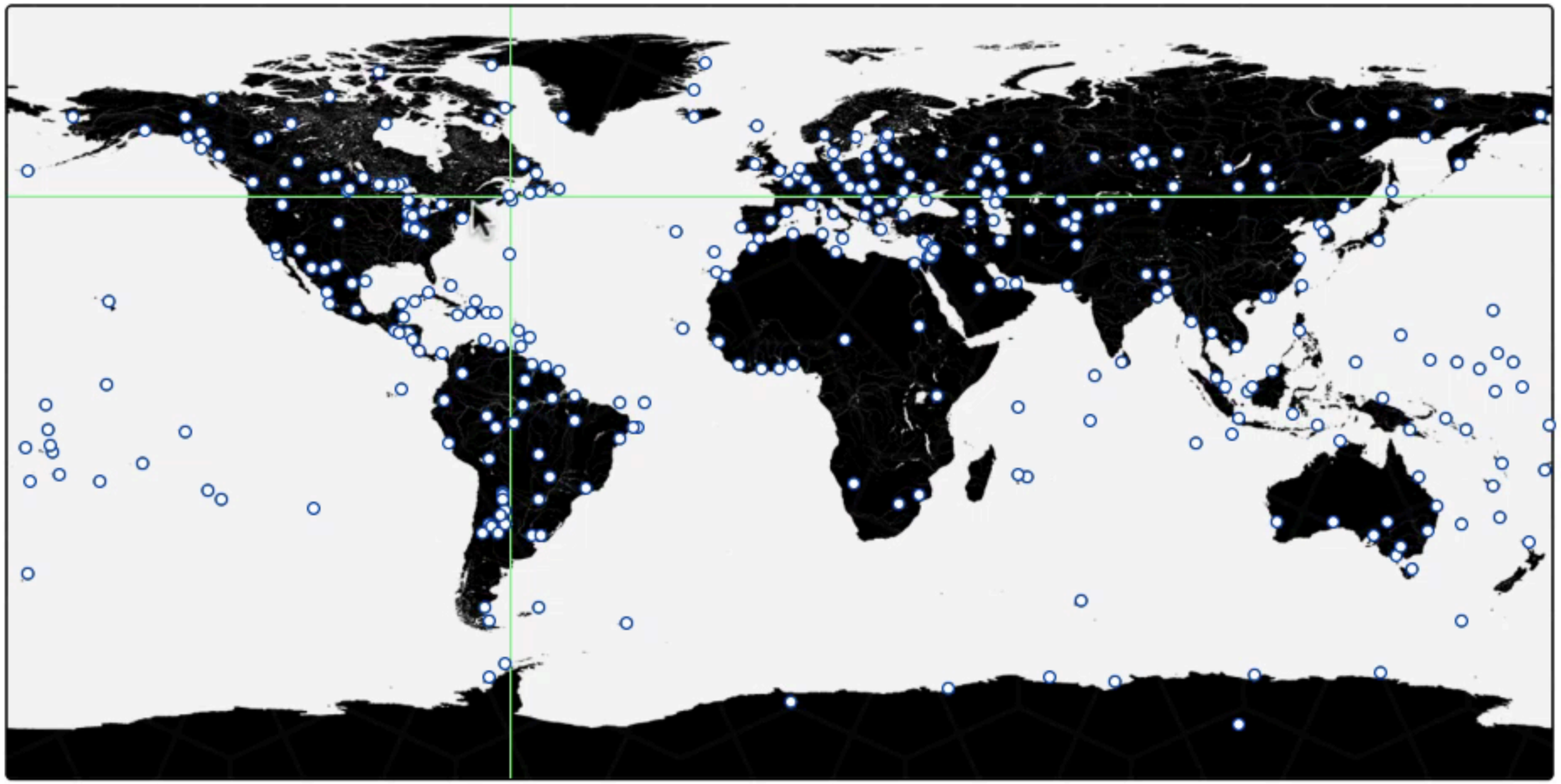
Timezone (Continent/City)

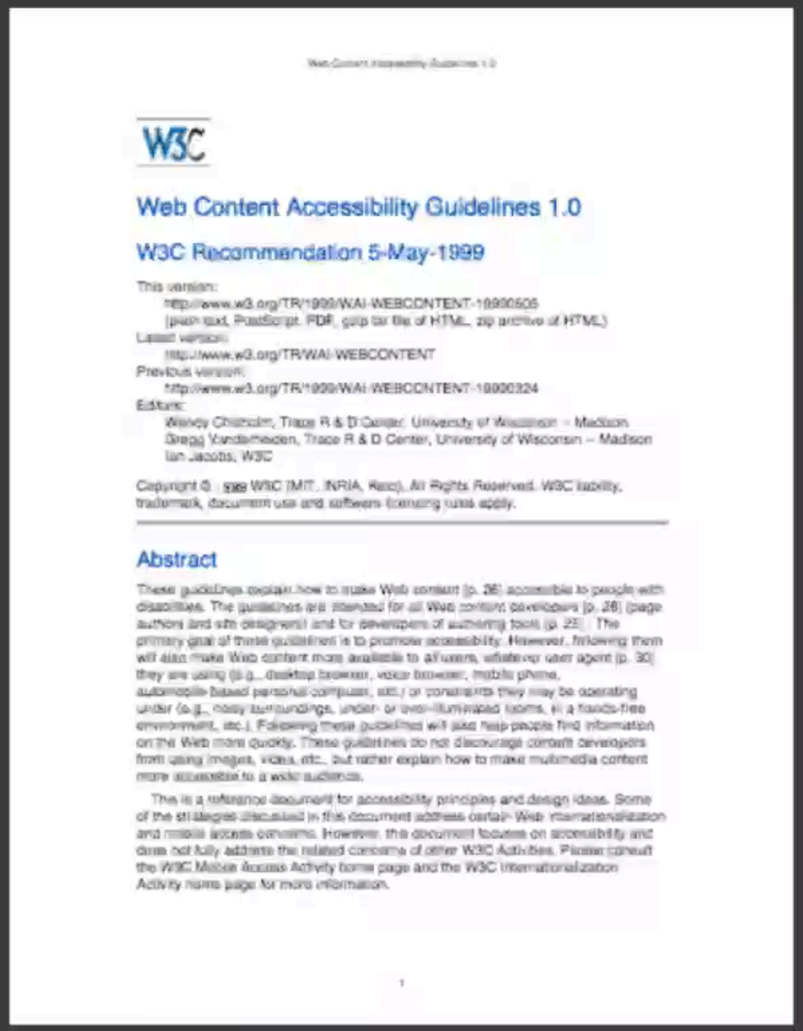
America/Moncton

```
  <style is="custom
  #timezone {
    --paper-timezon
    --paper-timezon
    --paper-timezon
    --paper-timezon
    --paper-timezon
  }
</style>
  <paper-timezone i
```

Timezone with format-time
hours format. For example

America/Moncton GMT -4.0 (01:26 pm AST)





Web Content Accessibility Guidelines 1.0

W3C Recommendation 5-May-1999

This version:
http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990505
(plain text, PostScript, PDF, gzip tar file of HTML, zip archive of HTML)
Latest version:
http://www.w3.org/TR/WAI-WEBCONTENT
Previous version:
http://www.w3.org/TR/1999/WAI-WEBCONTENT-19990324
Editors:
Wendy Chisholm, Trace R & D Center, University of Wisconsin – Madison
Gregg Vanderheiden, Trace R & D Center, University of Wisconsin – Madison
Ian Jacobs, W3C

Copyright © 1999 W3C (MIT, INRIA, Keio), All Rights Reserved. W3C liability, trademark, document use and software licensing rules apply.

Abstract

These guidelines explain how to make Web content [p. 26] accessible to people with disabilities. The guidelines are intended for all Web content developers [p. 26] (page authors and site designers) and for developers of authoring tools [p. 23]. The primary goal of these guidelines is to promote accessibility. However, following them will also make Web content more available to all users, whatever user agent [p. 30] they are using (e.g., desktop browser, voice browser, mobile phone, automobile-based personal computer, etc.) or constraints they may be operating under (e.g., noisy surroundings, under- or over-illuminated rooms, in a hands-free environment, etc.). Following these guidelines will also help people find information on the Web more quickly. These guidelines do not discourage content developers from using images, video, etc., but rather explain how to make multimedia content more accessible to a wide audience.

This is a reference document for accessibility principles and design ideas. Some of the strategies discussed in this document address certain Web internationalization and mobile access concerns. However, this document focuses on accessibility and does not fully address the related concerns of other W3C Activities. Please consult the W3C Mobile Access Activity home page and the W3C Internationalization Activity home page for more information.

Choose your newspaper

Web Content Accessibility Guidelines 1.0

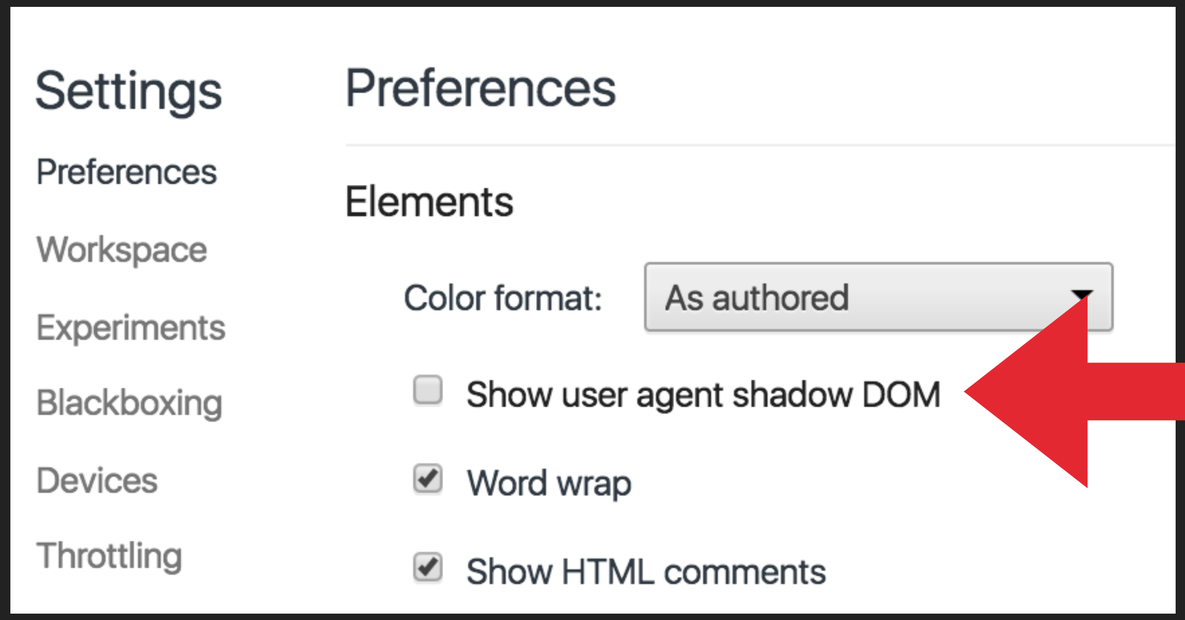
Single page

Double page

```
<pdf-viewer id="viewer" initial-zoom="fit"
  src="fichier.pdf" page="..." pages="..."
  mode="..."></pdf-viewer>
```

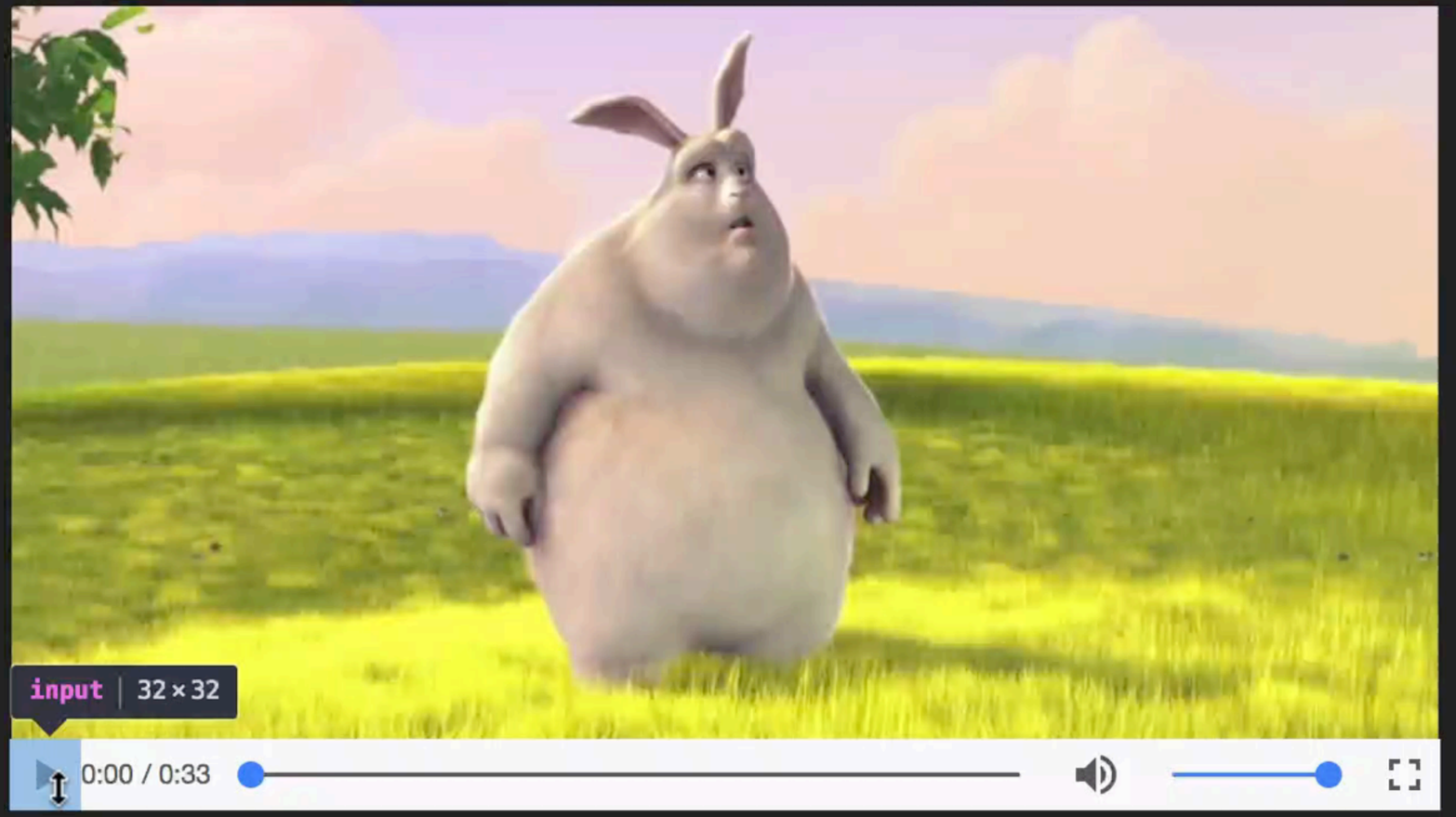
RÉVÉLER L'INVISIBLE DES COMPOSANTS NATIFS

(dans Chrome)



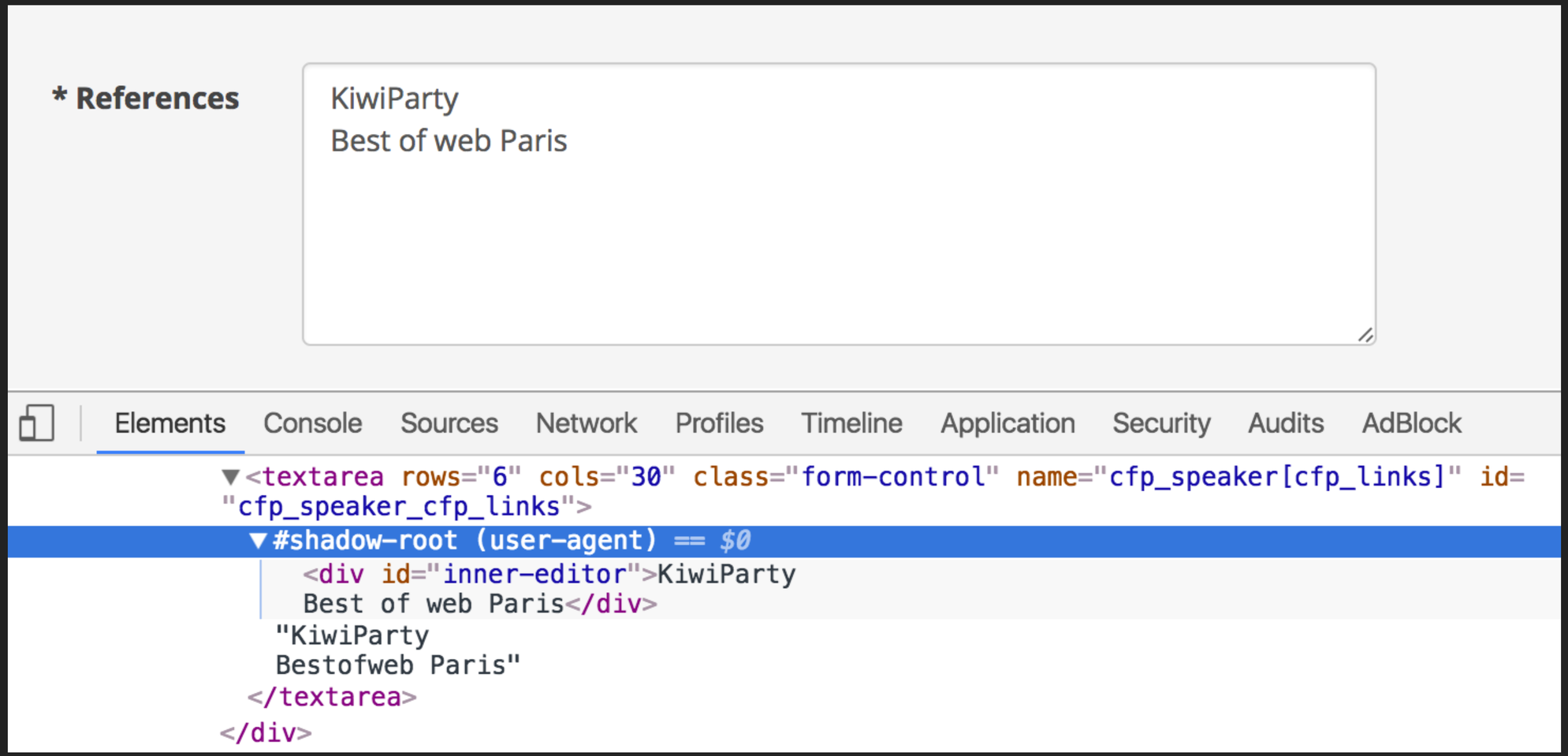
LES WEB COMPONENTS SONT LES MÊMES OUTILS QUE CEUX DES DÉVELOPPEURS DE NAVIGATEUR

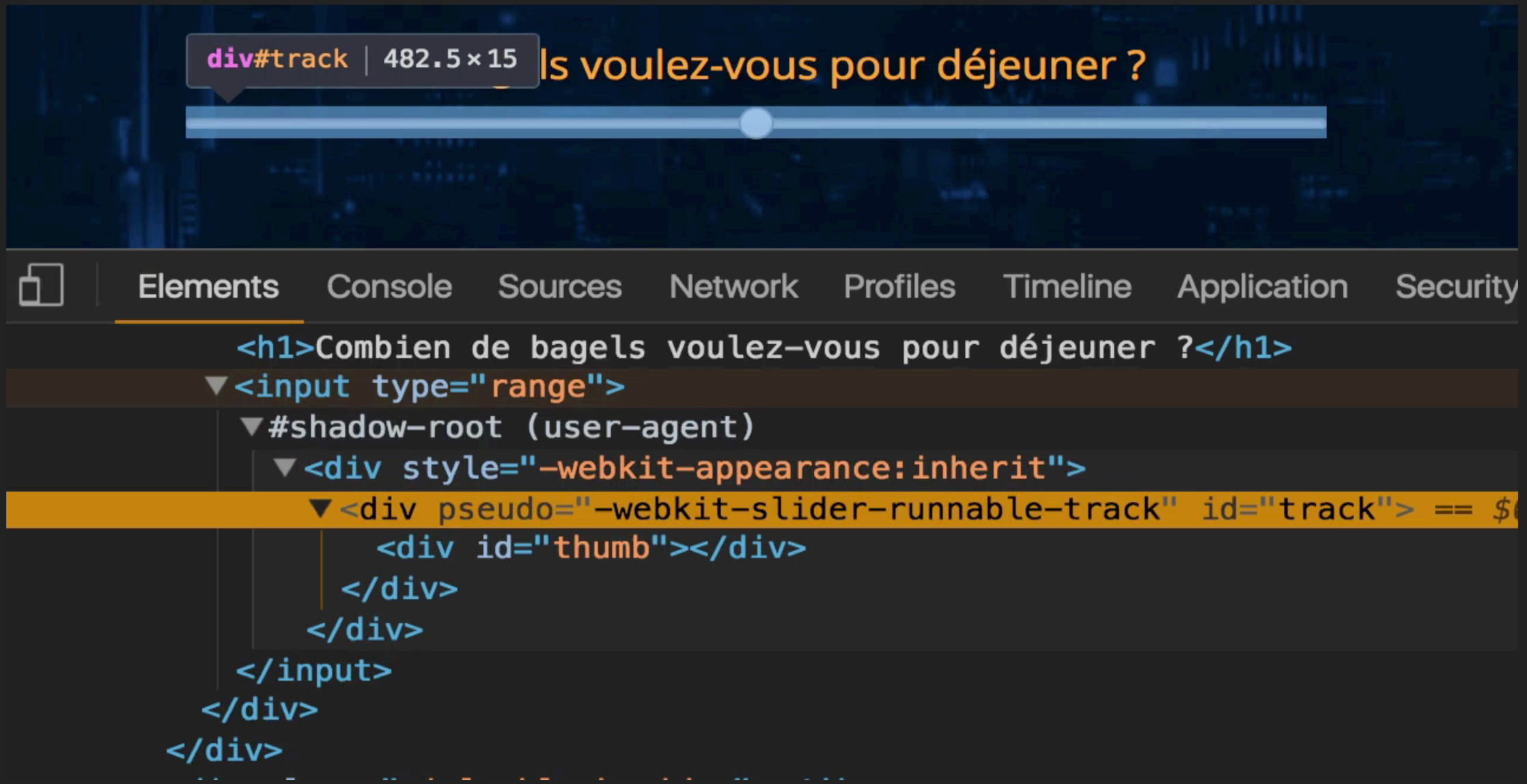
(et c'est bien)



Elements Console Sources Network Profiles Timeline Application Security Audits AdBlock

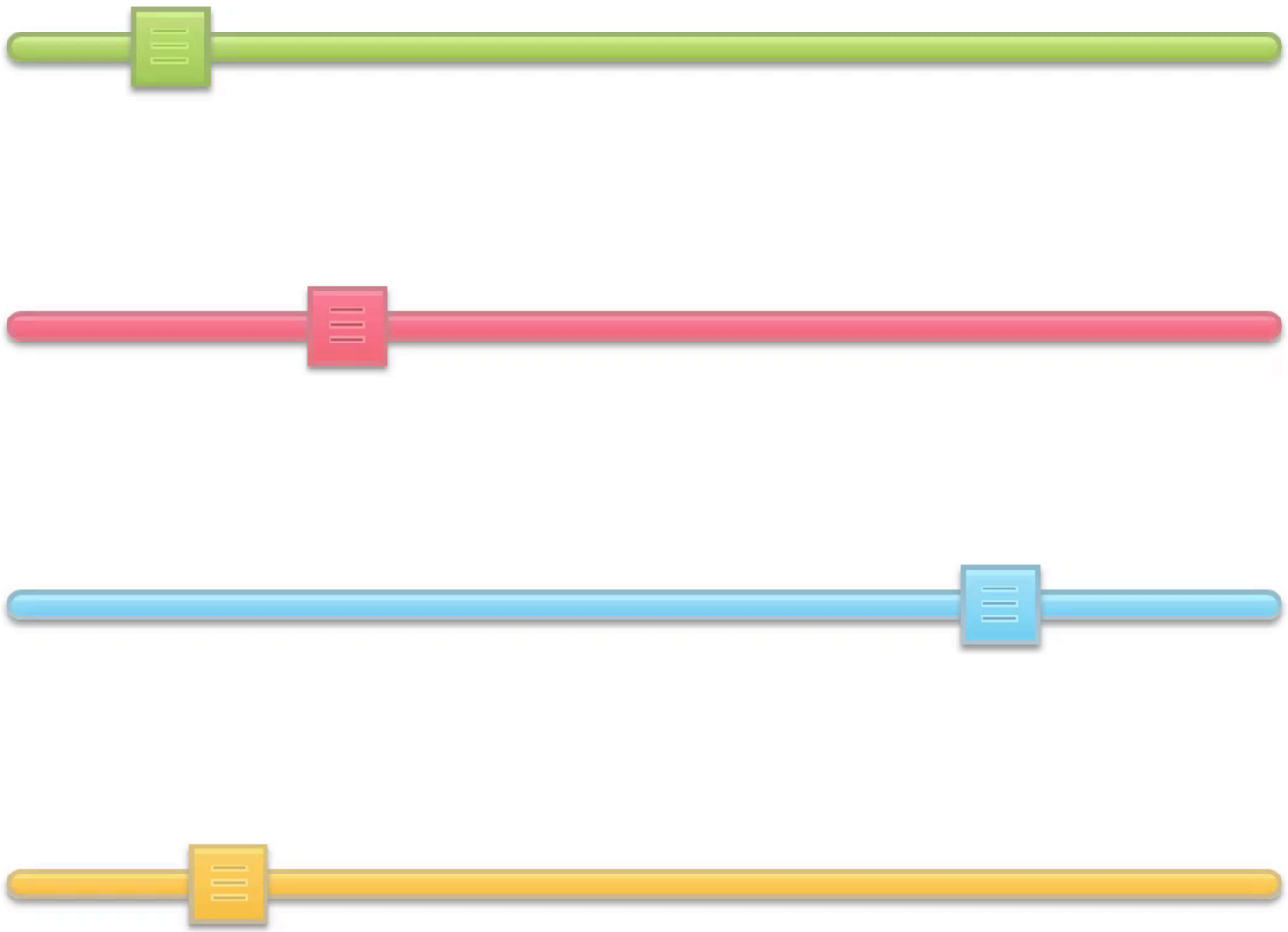
```
▼#shadow-root (user-agent)
  ▼<div pseudo="-webkit-media-controls"> == $0
    ▶<div pseudo="-webkit-media-controls-overlay-enclosure">…</div>
    ▼<div pseudo="-webkit-media-controls-enclosure">
      ▼<div pseudo="-webkit-media-controls-panel" style>
        ▶<input type="button" pseudo="-webkit-media-controls-play-button" style>…</input>
         <div pseudo="-webkit-media-controls-current-time-display" style>0:00</div>
         <div pseudo="-webkit-media-controls-time-remaining-display" style>/ 0:33</div>
        ▶<input type="range" step="any" pseudo="-webkit-media-controls-timeline" max="33.042" styl
        ▶<input type="button" pseudo="-webkit-media-controls-mute-button" style>…</input>
        ▶<input type="range" step="any" max="1" pseudo="-webkit-media-controls-volume-slider" styl
```


* References
KiwiParty
Best of web Paris
Elements
Console
Sources
Network
Profiles
Timeline
Application
Security
Audits
AdBlock
▼<textarea rows="6" cols="30" class="form-control" name="cfp_speaker[cfp_links]" id="cfp_speaker_cfp_links">
▼#shadow-root (user-agent) == $0
KiwiParty
Best of web Paris
"KiwiParty
Bestofweb Paris"
</textarea>


div#track | 482.5 × 15
ls voulez-vous pour déjeuner ?
Elements
Console
Sources
Network
Profiles
Timeline
Application
Security
<h1>Combien de bagels voulez-vous pour déjeuner ?</h1>
<input type="range">
#shadow-root (user-agent)

 == $

</input>



http://codepen.io/thebabydino/

HEIN ?

COMMENT ÇA MARCHE ?

Shadow host

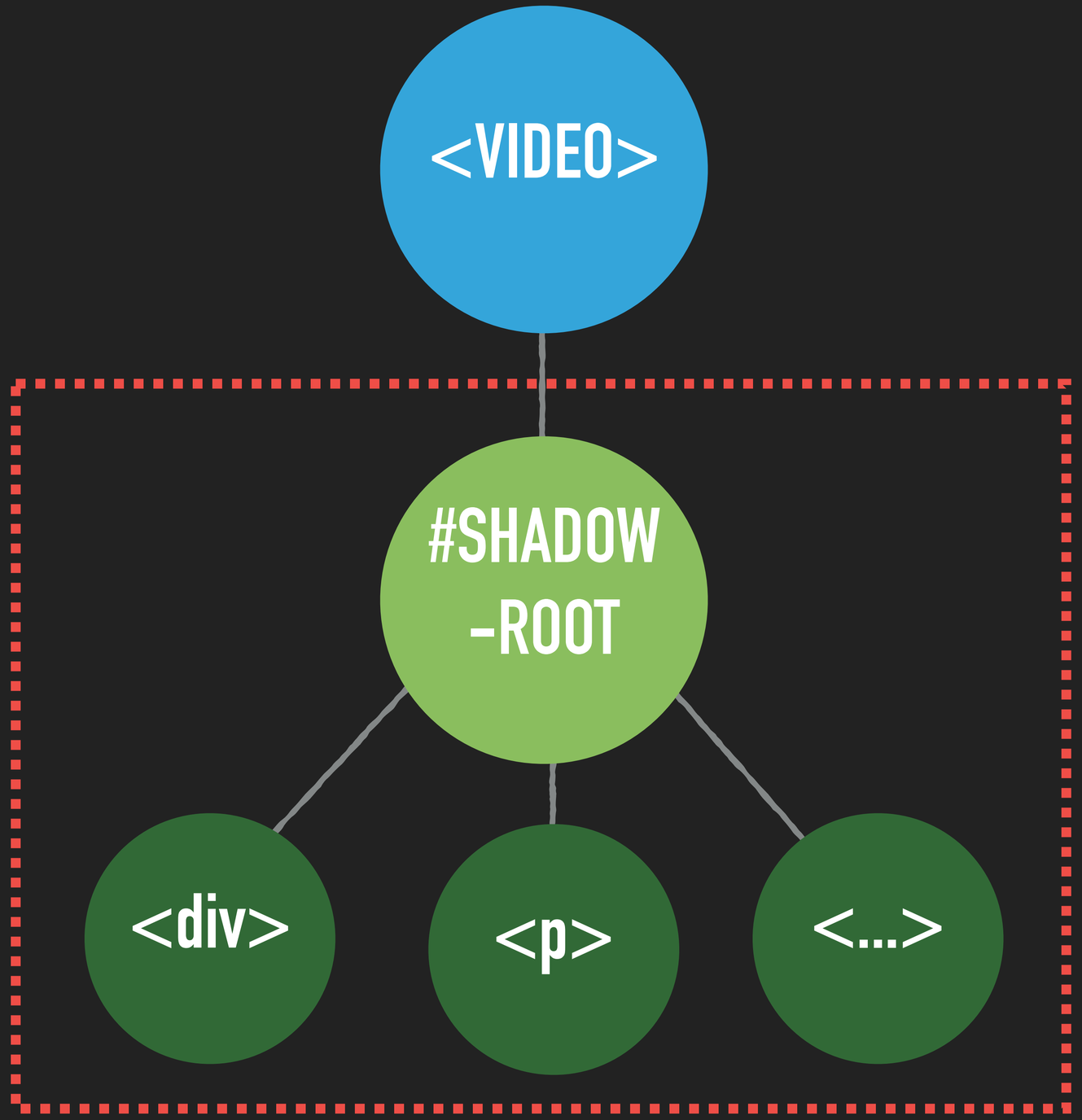
noeud de la page
qui contient tout

Shadow root

premier noeud
dans le shadow dom

Shadow boundary

barrière protectrice



Shadow host

noeud de la page
qui contient tout

Shadow root

premier noeud
dans le shadow dom

Shadow boundary

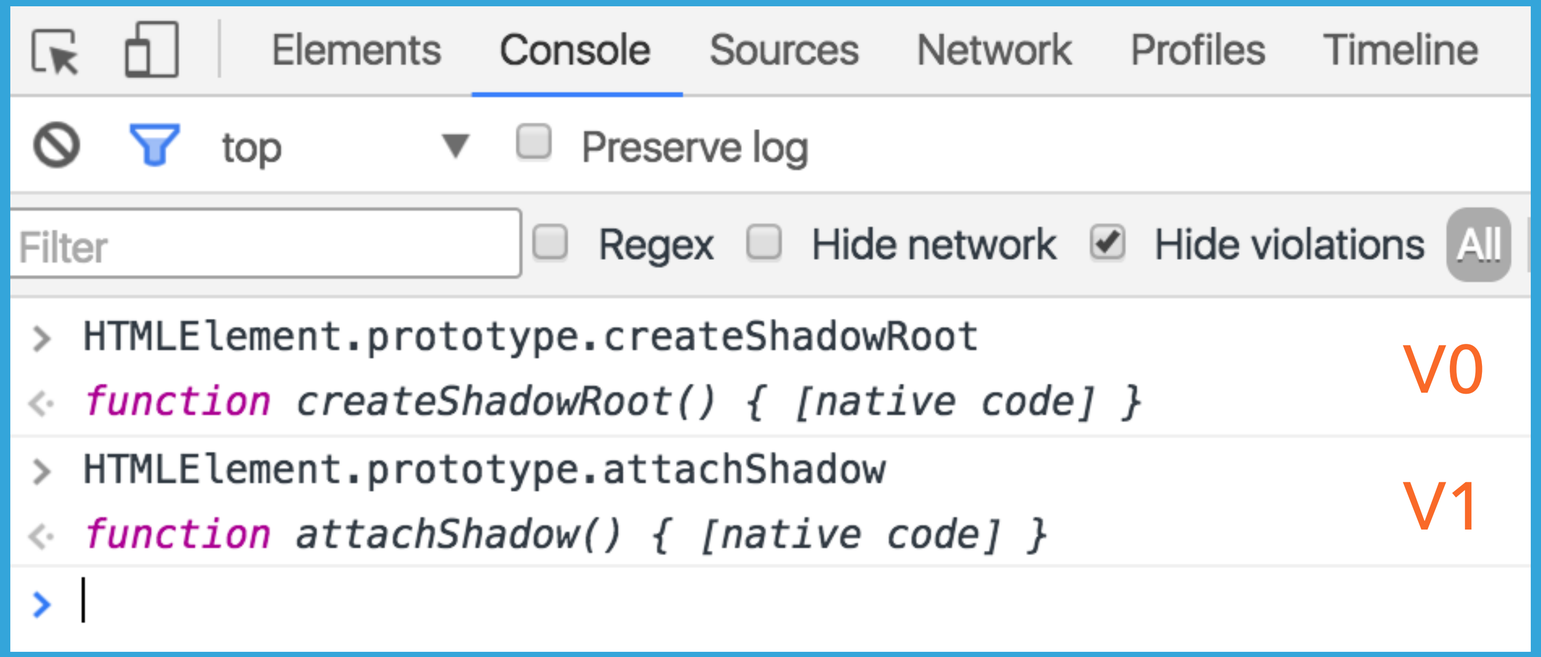
barrière protectrice

Elements Console Sources Network Profiles Timeline Application Security Audits AdBlock

```
<video id="mavideo" width="720" height="406" poster="bunny.jpg">
  #shadow-root (user-agent) == $0
    <div pseudo="-webkit-media-controls">
      <div pseudo="-webkit-media-controls-overlay-enclosure">…</div>
      <div pseudo="-webkit-media-controls-enclosure">…</div>
      <div pseudo="-internal-media-controls-text-track-list" style="display: none;"></div>
      <div pseudo="-internal-media-controls-overflow-menu-list" style="display: none;">…</div>
    </div>
  <source src="bunny.mp4" type="video/mp4">
  <source src="bunny.ogv" type="video/ogg">
  <source src="bunny.webm" type="video/webm">
</video>
</body>
```

TESTER LE SUPPORT

```
const supportsShadowDOMV1 =
!!HTMLElement.prototype.attachShadow;
```

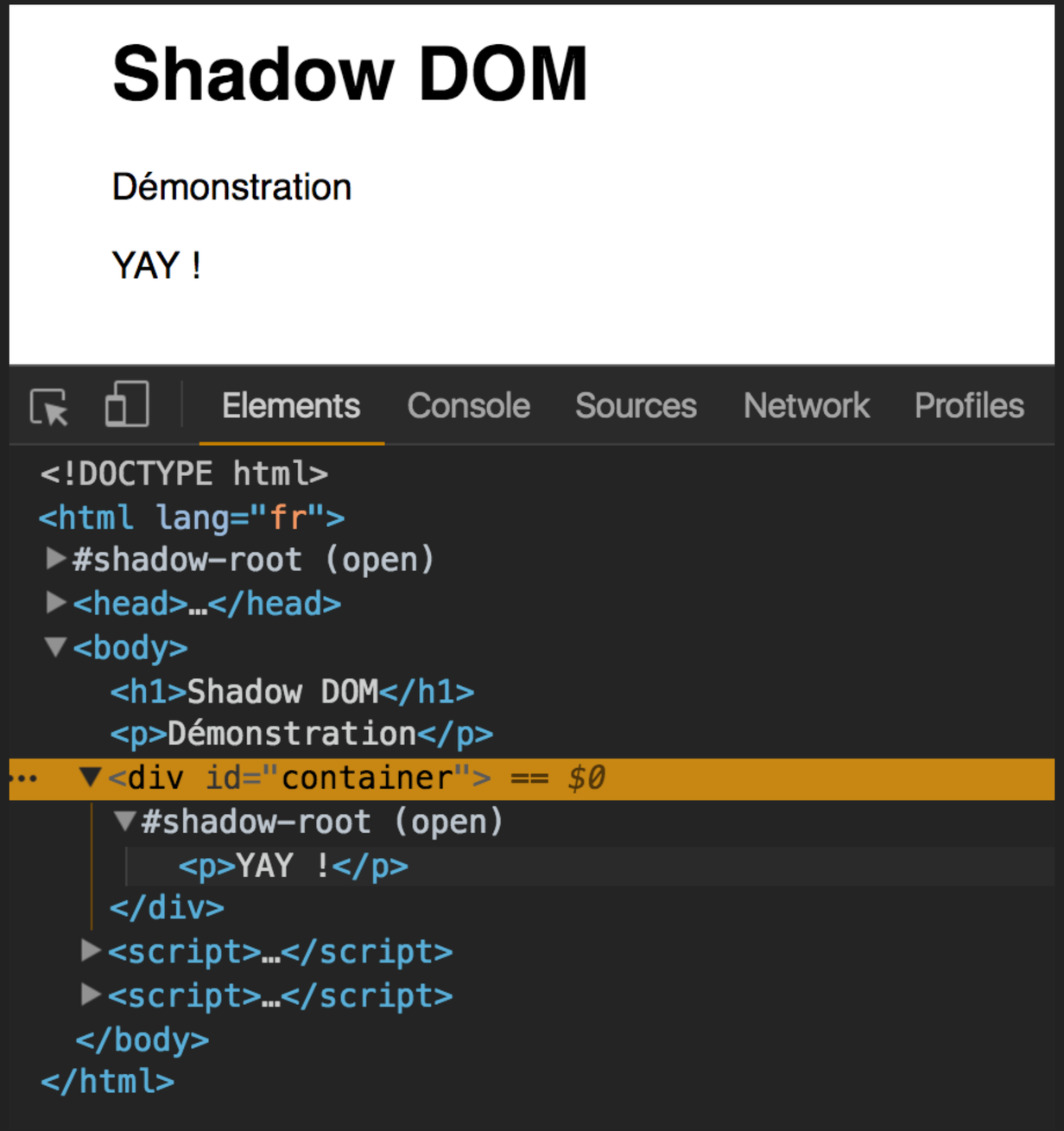


TRANSFORMER UN ÉLÉMENT EN SHADOW HOST

```
Element.attachShadow({mode: 'open'});
```

si mode vaut 'closed' le JavaScript externe est bloqué

```
<h1>Shadow DOM</h1>
<p>Démonstration</p>
<div id="container"></div>

<script>
var container = document.querySelector('#container');
var root = container.attachShadow({mode: 'open'});
root.innerHTML = '<p>YAY !</p>';
</script>
```


Shadow DOM
Démonstration
YAY !
Elements
Console
Sources
Network
Profiles
<!DOCTYPE html>
<html lang="fr">
#shadow-root (open)
<head>…</head>
<body>
<h1>Shadow DOM</h1>
Démonstration
 == $0
#shadow-root (open)
YAY !

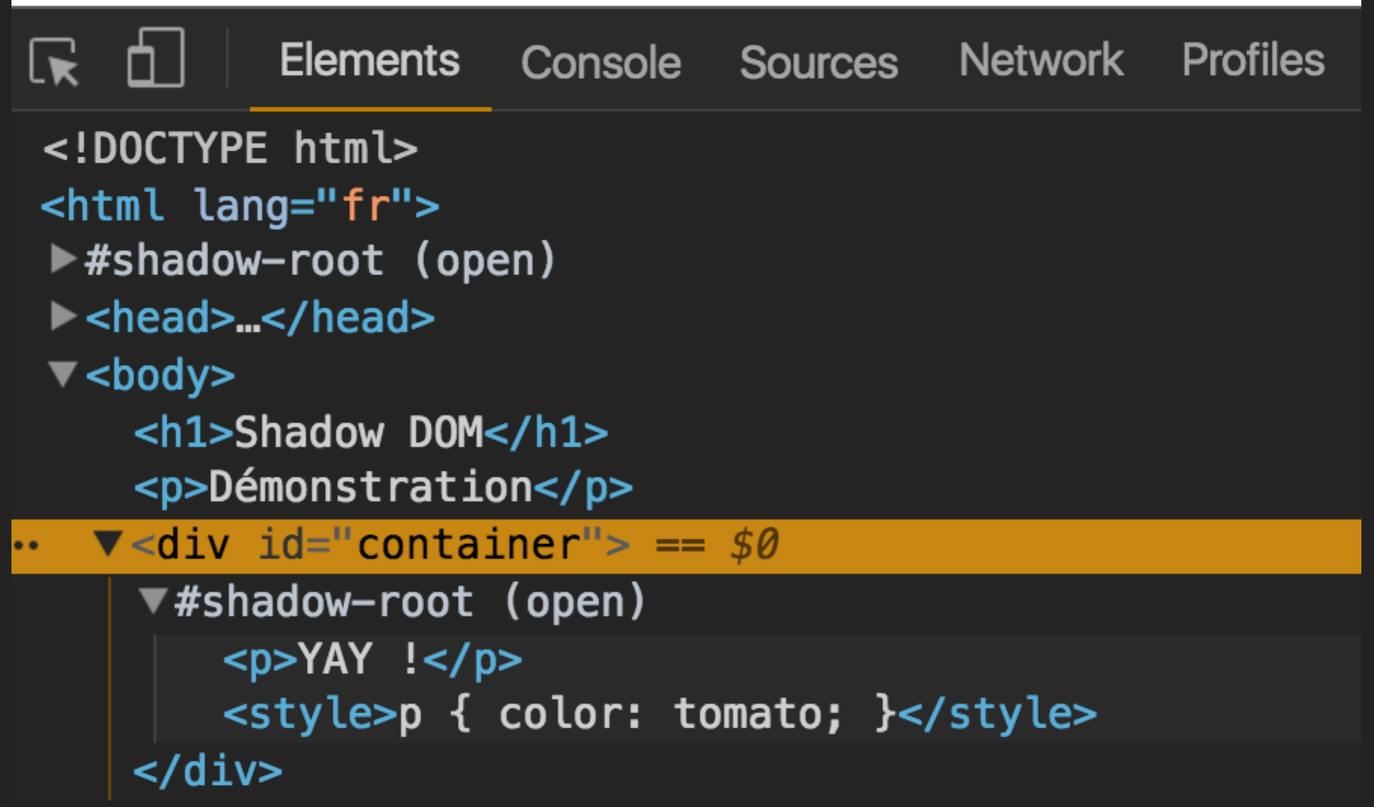
<script>…</script>
<script>…</script>
</body>
</html>

CSS

```
root.innerHTML += '<style>p { color: tomato; }</style>';
```

n'affecte que les contenus du Shadow DOM

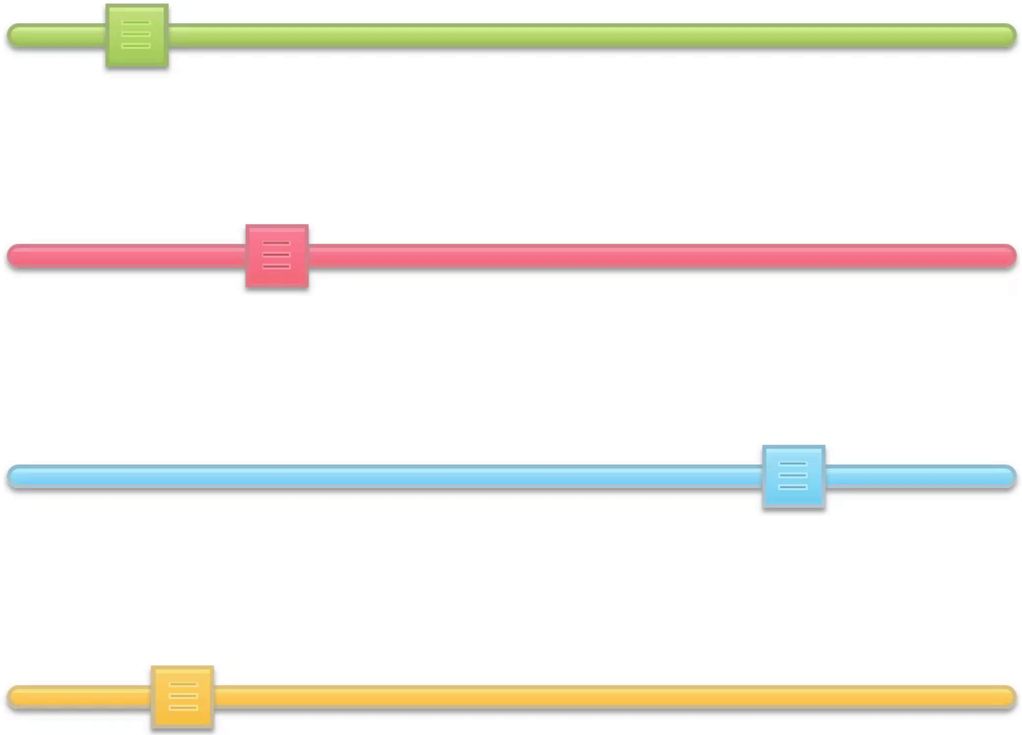
SEUL CE PARAGRAPHE EST STYLÉ



inversement, le contenu du Shadow DOM
n'est pas affecté par les styles du document parent

```
#container p {
  background: green !important;
  /* Ne fonctionne pas */
}
```

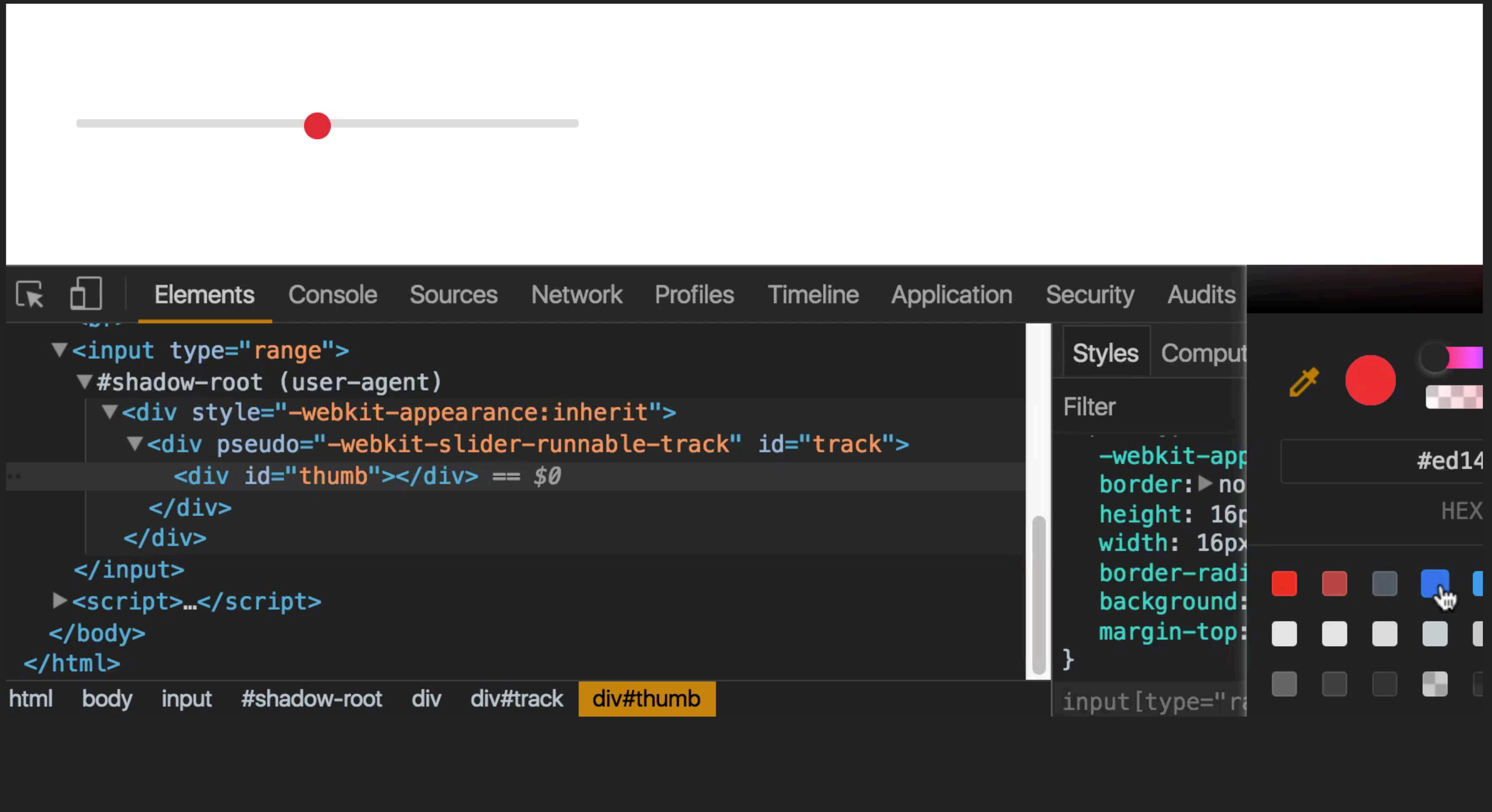
mais alors comment
styler les éléments natifs
tels que
<video> ou <input
type="range">
si on n'arrive pas à cibler leur
contenu ?



avec les pseudo-éléments

```
::-webkit-slider-runnable-track
::-webkit-slider-thumb
:focus
:focus::-webkit-slider-runnable-track
::-moz-range-track
::-moz-range-thumb
:-moz-focusring
::-ms-track
::-ms-fill-lower
::-ms-fill-upper
::-ms-thumb
:focus::-ms-fill-lower
:focus::-ms-fill-upper
```

http://brennaobrien.com/blog/2014/05/style-input-type-range-in-every-browser.html


Elements
Console
Sources
Network
Profiles
Timeline
Application
Security
Audits
<input type="range">
#shadow-root (user-agent)

 == $0

</input>
<script>…</script>
</body>
</html>
Styles
Filter
-webkit-app
border:
height: 16p
width: 16px
border-radi
background:
margin-top:
}
input[type="r
#ed14
HEX
html
body
input
#shadow-root
div
div#track
div#thumb

```
<style>:host { background: royalblue; }</style>
```

styler l'élément shadow host (conteneur)
depuis l'intérieur du shadow DOM

```
:host(.theme1) {
  background: hotpink;
}
:host(.theme2) {
  background: olivedrab;
}

:host(:hover) {
  background:red;
}
```

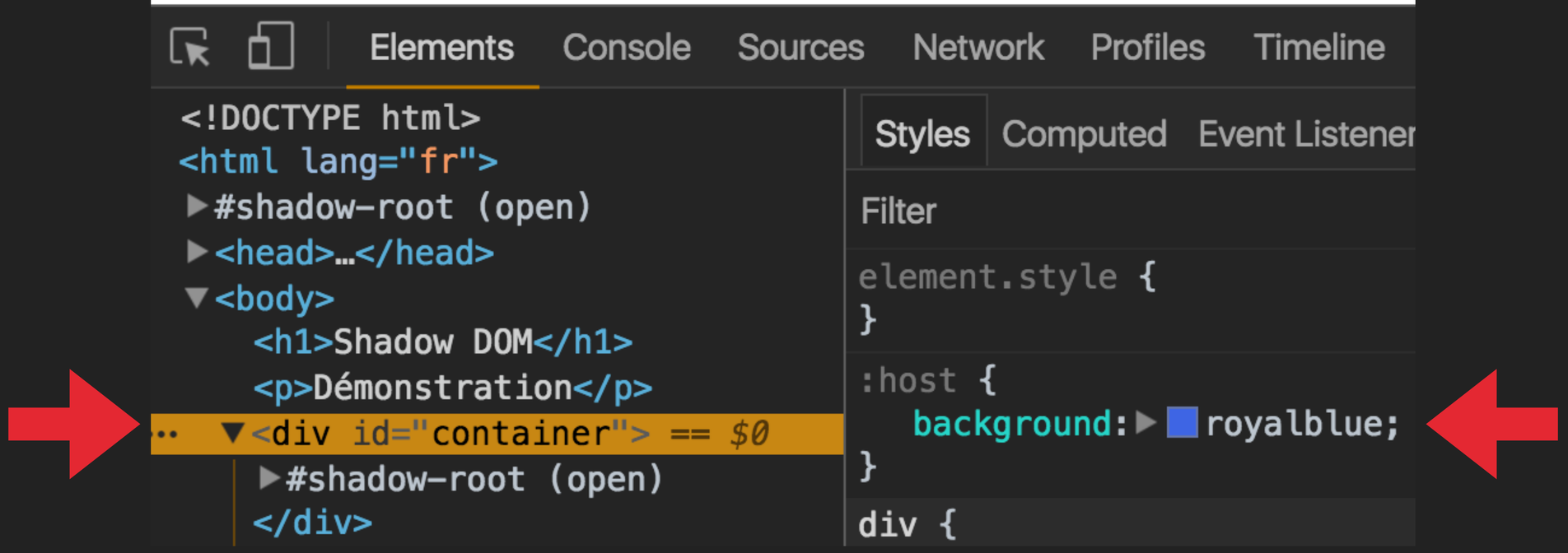
styler l'élément shadow host (conteneur)
en fonction de sa classe ou de son état

Shadow DOM

Démonstration

YAY !

Un bouton



ÉVÉNEMENTS JAVASCRIPT

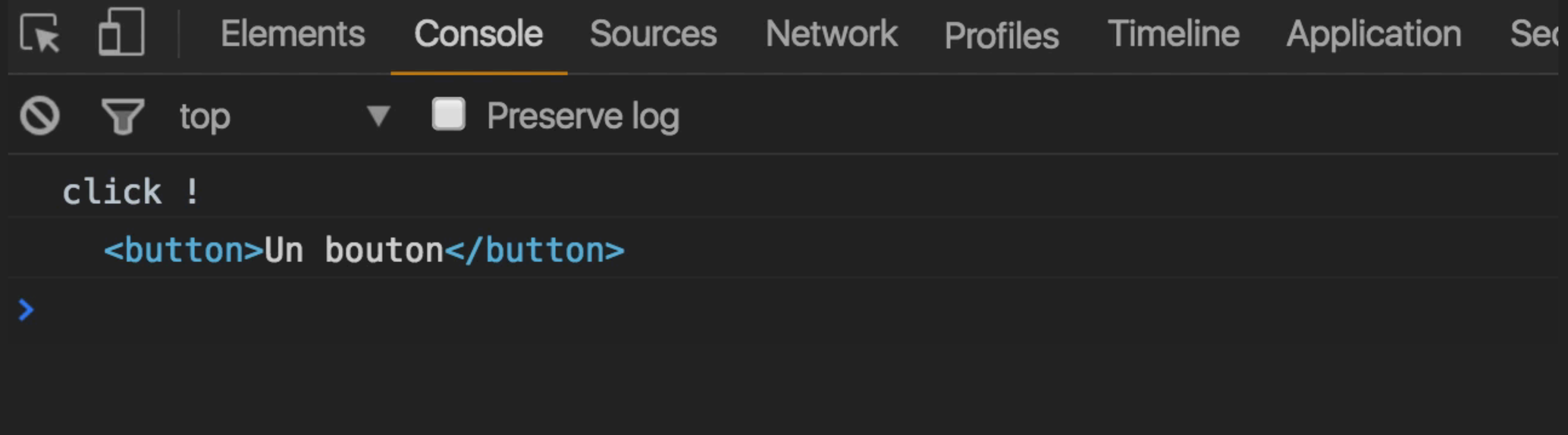
```
root.innerHTML += '<button>Un bouton</button>';

var button = root.querySelector('button');
button.addEventListener('click',function(e) {
  console.log('click !');
  console.log(e.target);
});
```

Shadow DOM

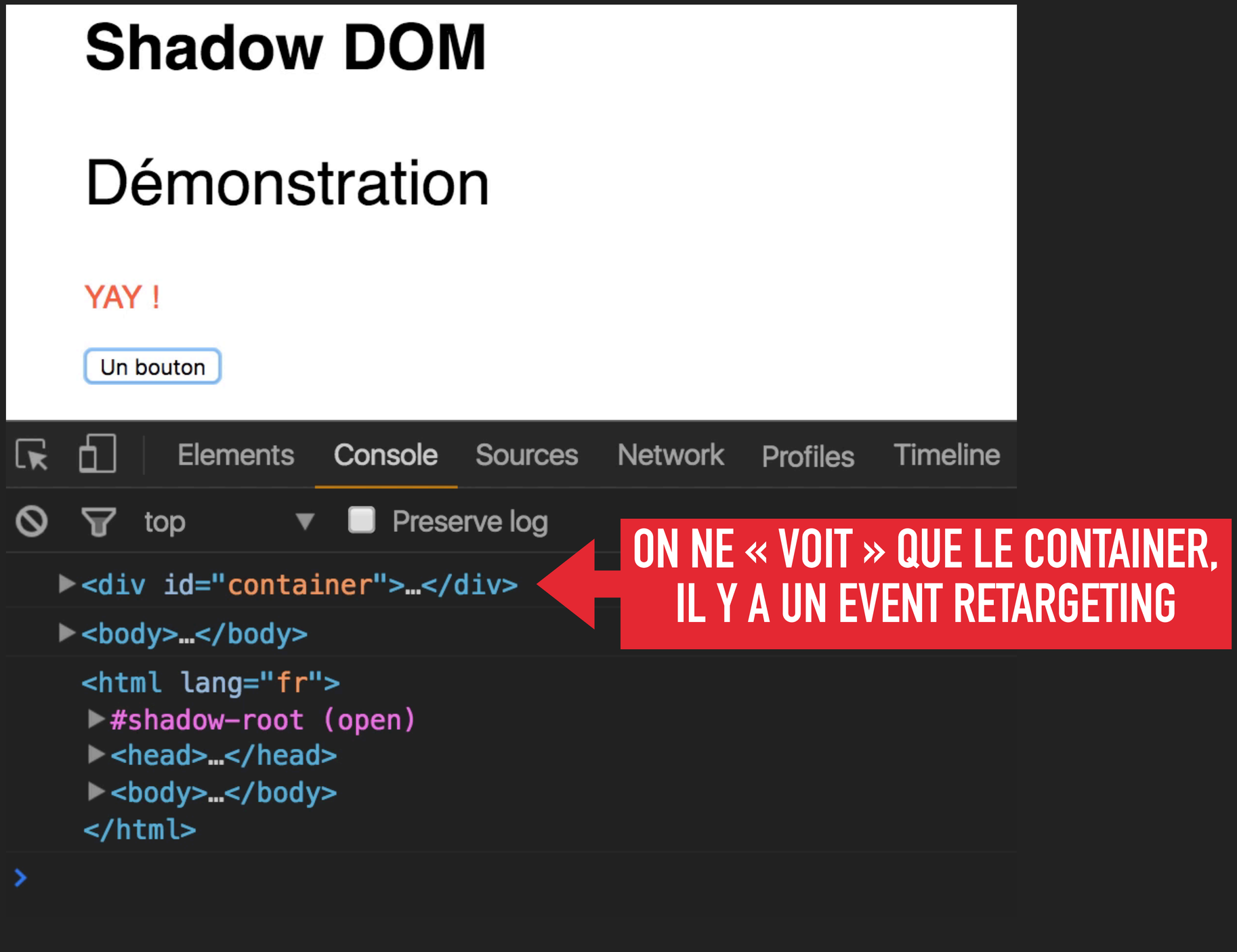
Démonstration

YAY !

Un bouton



```
// Quels éléments de la page reçoivent un clic ?
document.querySelectorAll('*').forEach(function(el) {
  el.addEventListener('click',function(e) {
    console.log(el);
  });
});
```


Shadow DOM
Démonstration
YAY !
Un bouton
Elements
Console
Sources
Network
Profiles
Timeline
top
Preserve log
…
<body>…</body>
<html lang="fr">
#shadow-root (open)
<head>…</head>
<body>…</body>
</html>
ON NE « VOIT » QUE LE CONTAINER,
IL Y A UN EVENT RETARGETING

ATTENTION
MOMENT CODE
ÇA VA ÊTRE CHAUD

<SLOT>

+

<TEMPLATE>

+

CUSTOM ELEMENTS

<SLOT>

une manière de faire de la composition,
du templating (prévoir des emplacements à remplir)

```
<div id="container">
▼#shadow-root (open)
    <span>Conférence :</span>
  ▼<slot name="conf">
    </slot>
    <span>Ville :</span>
  ▶<slot name="city">…</slot>
    <img slot="photo">
</div>
```

<TEMPLATE>

contient du code **inerte** qui sera utile plus tard :
scripts non exécutés, images non chargées, html non interprété

```
<template class="city">
  <style>p { color: white; padding: 1em; margin: 0; }
    .ville { background: rebeccapurple; }
    .province { background: crimson; margin-bottom:1em; }</style>
  <p class="ville">Ville : <slot name="city"></slot><br>
  <p class="province">Province : <slot name="province"></slot></p>
</template>
```

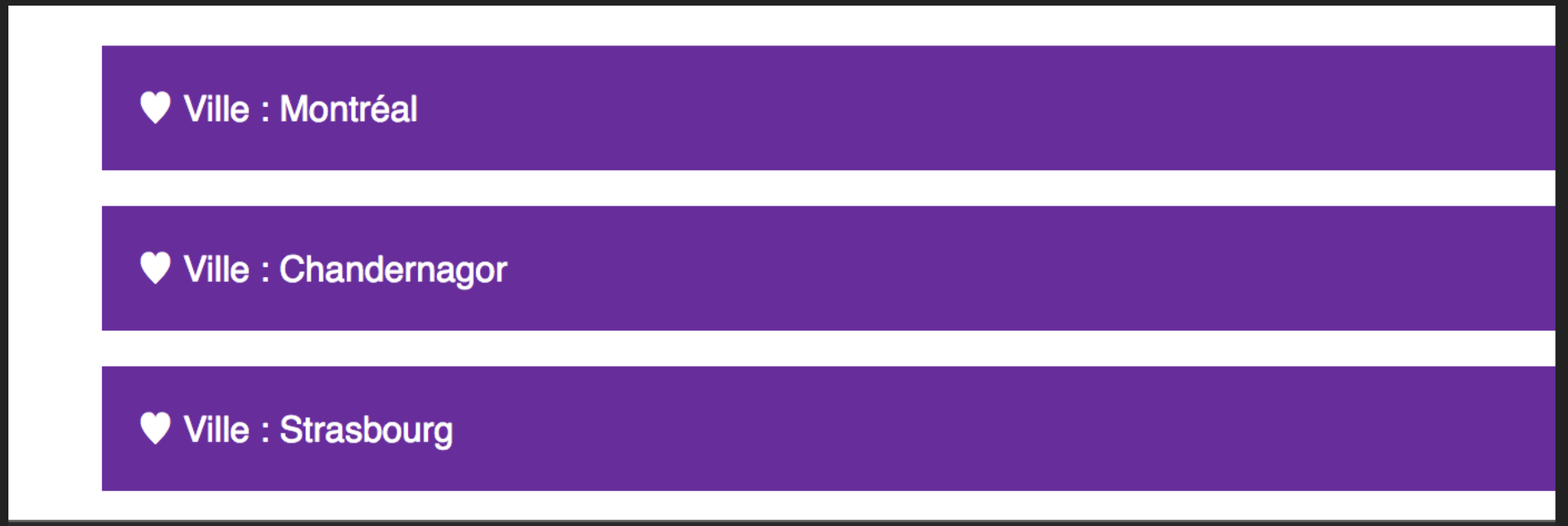
SLOTS

CUSTOM ELEMENTS

```
<script>
customElements.define('x-confoo', class extends HTMLElement {
  constructor() {
    super();
    var root = this.attachShadow({mode: 'open'});
    var tpl = document.querySelector('.city');
    root.appendChild(tpl.content.cloneNode(true));
  }
});
</script>
```

ON POURRA ÉCRIRE <x-confoo></x-confoo>

```
<template class="city">
  <style>p { background: rebeccapurple; color: white; padding: 1em; }</style>
  <p><span aria-hidden="true">&#9829;</span>
    Ville : <slot></slot></p>
</template>

<script>
customElements.define('x-confoo', class extends HTMLElement {
  constructor() {
    super();
    var root = this.attachShadow({mode: 'open'});
    var tpl = document.querySelector('.city');
    root.appendChild(tpl.content.cloneNode(true));
  }
});
</script>

<x-confoo>Montréal</x-confoo>
<x-confoo>Chandernagor</x-confoo>
<x-confoo>Strasbourg</x-confoo>
```



Elements Console Sources Network Profiles Timeline Application Security

```
▶<template class="city">…</template> == $0
▶<script>…</script>
▼<x-confoo>
  ▼#shadow-root (open)
    <style>p { background: rebeccapurple; color: white; padding: 1em; }</style>
    ▶<p>…</p>
  "Montréal"
</x-confoo>
▼<x-confoo>
  ▶#shadow-root (open)
  "Chandernagor"
</x-confoo>
▶<x-confoo>…</x-confoo>
```

```
<template class="city">
  <style>p { color: white; padding: 1em; margin: 0; }
    .ville { background: rebeccapurple; }
    .province { background: crimson; margin-bottom:1em; }</style>
  <p class="ville">Ville : <slot name="city"></slot><br>
  <p class="province">Province : <slot name="province"></slot></p>
</template>

<script>
customElements.define('x-confoo', class extends HTMLElement {
  constructor() {
    super();
    var root = this.attachShadow({mode: 'open'});
    var tpl = document.querySelector('.city');
    root.appendChild(tpl.content.cloneNode(true));
  }
});
</script>

<x-confoo><span slot="city">Montréal</span><span slot="province">Québec</span></x-confoo>
<x-confoo><span slot="city">Vancouver</span><span slot="province">Colombie-Britannique</span></x-confoo>
```

Elements Console Sources Network Profiles Timeline

```
▼<x-confoo>
  ▼#shadow-root (open)
    ▶<style>…</style>
    ▼<p class="ville">
       "Ville : "
      ▼<slot name="city">
         ↳ <span>
       </slot>
       <br>
     </p>
    ▶<p class="province">…</p>
  <span slot="city">Montréal</span> == $0
  <span slot="province">Québec</span>
</x-confoo>
▶<x-confoo>…</x-confoo>
```

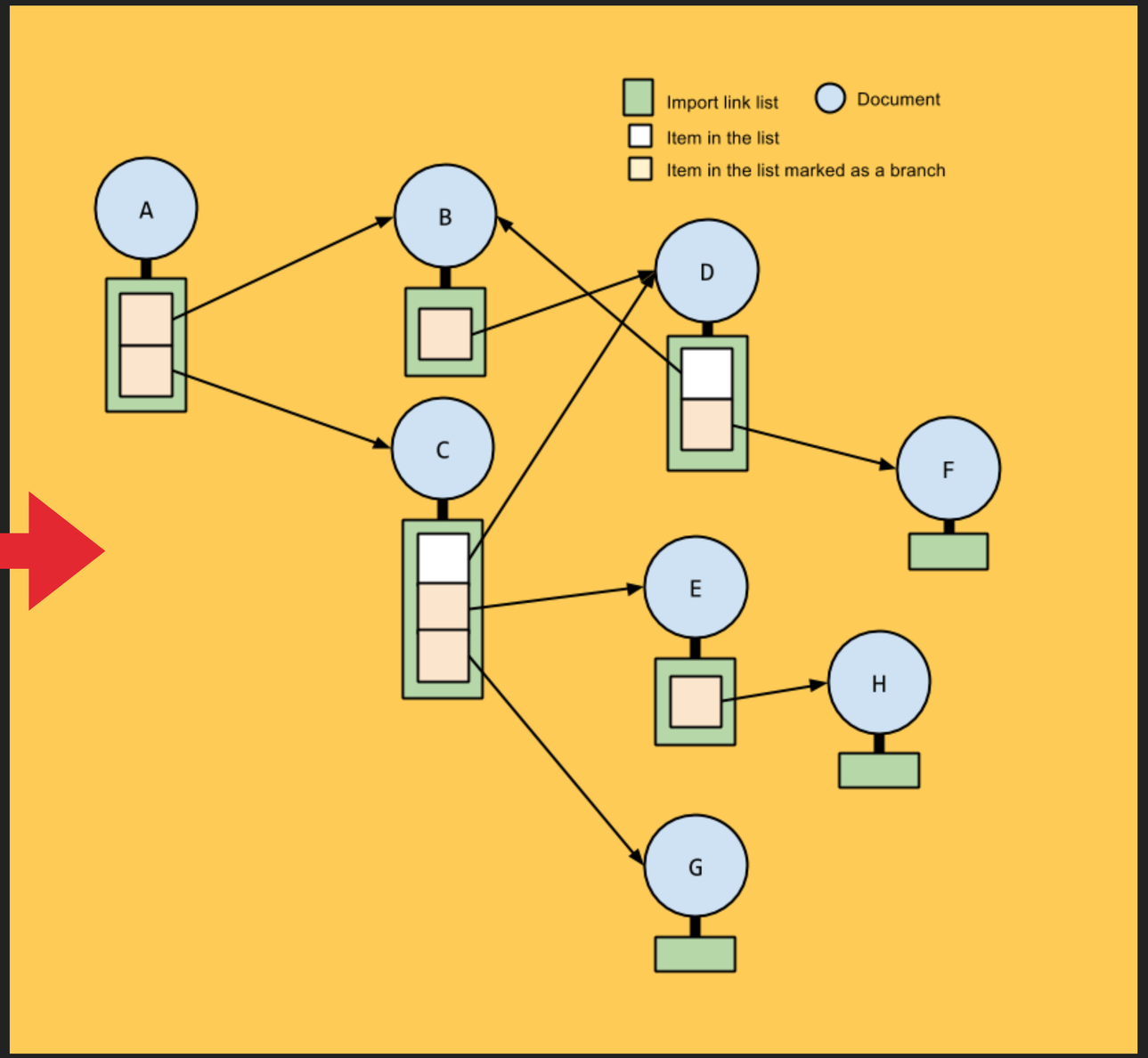
- CUSTOM ELEMENTS = <TEMPLATE> + SHADOW DOM
  + <SLOT>

HTML IMPORTS

https://www.w3.org/TR/html-imports/

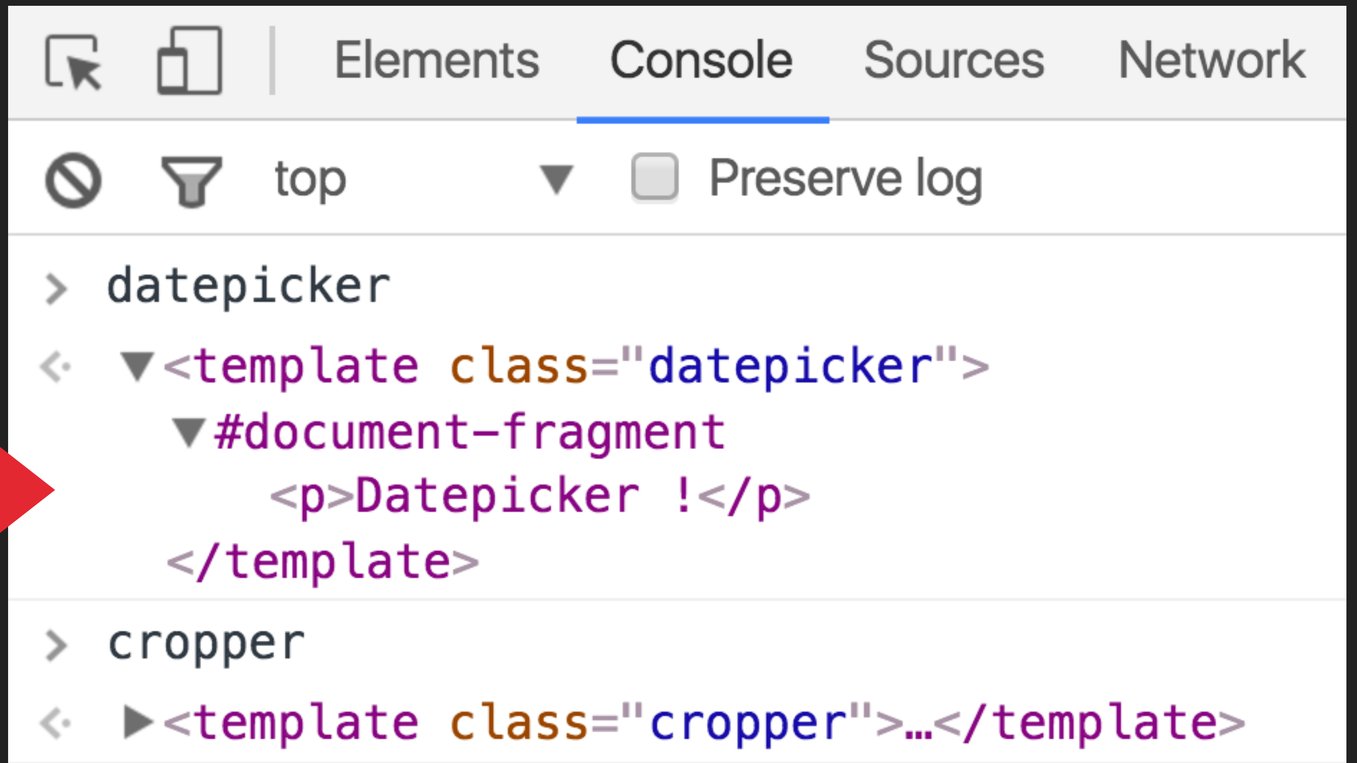
(supporté uniquement par Chrome/Opera ; Edge « Under Consideration »)

ÇA FAIT PEUR


Import link list
Document
Item in the list
Item in the list marked as a branch
A
B
C
D
E
F
G
H

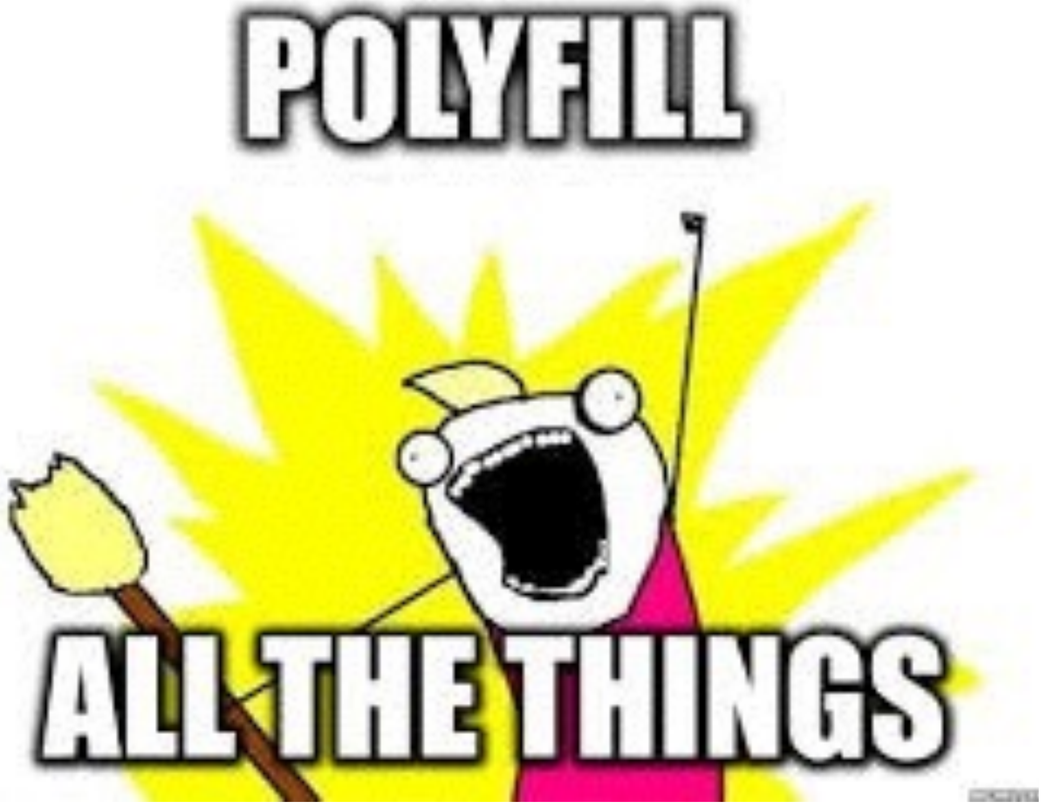
```
<link rel="import" id="foo" href="/test/composants.html">

<script>
var contents = document.querySelector('#foo').import;
var datepicker = contents.querySelector('.datepicker');
var cropper = contents.querySelector('.cropper');
</script>
```

ON IMPORTE DES FRAGMENTS DE DOM DEPUIS UN AUTRE DOCUMENT



ON RÉSUME LES WEB COMPONENTS

- SHADOW DOM = ENCAPSULATION
- TEMPLATES = BOUTS DE DOM
- CUSTOM ELEMENTS = EXTENSIONS
- HTML IMPORTS = « PACKAGING »



AVEC WEBCOMPONENTS ET POLYMER, ENTRE AUTRES

en plus polymer 2.0 est axé sur les spécifications natives (mais non finalisé, attention)

```
bower install --save webcomponents/webcomponentsjs
```

POLYFILL

IMPORTS

CUSTOM ELEMENTS

```
<!DOCTYPE html>
<html>
  <head>
    <!-- 1. Load webcomponents-lite.min.js for polyfill support. -->
    <script src="bower_components/webcomponentsjs/webcomponents-lite.min.js">
    </script>

    <!-- 2. Use an HTML Import to bring in some elements. -->
    <link rel="import" href="bower_components/paper-button/paper-button.html">
    <link rel="import" href="bower_components/paper-input/paper-input.html">
  </head>
  <body>
    <!-- 3. Declare the element. Configure using its attributes. -->
    <paper-input label="Your name here"></paper-input>
    <paper-button>Say Hello</paper-button>
    <div id="greeting"></div>
    ...
```

QUI A COPIÉ QUI ?

PRINCIPES ~SEMBLABLES

SEMBLABLES (QUOIQUE…)

- Flash
- Bibliothèques YUI, Dojo (hacks) ; React (Virtual DOM)
- iframes (dont iframes avec attribut srcdoc)
- SVG (notamment avec svg:use)
- Element Queries
- CSS Containment (premier Working Draft !)
  https://www.w3.org/TR/css-contain-1/

CONCLUSION

VERDICT

IL NE FAUT PAS SE LE CACHER : ÇA DONNE DES BOUTONS AUX INTÉGRATEURS…

C'EST UNE BONNE API QUI GÈRE LE SCOPE CSS, LE SCOPE DOM... ET SE COMBINE AVEC LES AUTRES API WEB.

AVEC CUSTOM ELEMENTS, SHADOW DOM PERMET DES VRAIS COMPOSANTS ENCAPSULÉS SANS HACK OU <IFRAME>

ACCESSIBILITÉ : C'EST PLUTÔT PAS MAL ! ET MIEUX QUE LES BOÎTES NOIRES DES PLUGINS, DE FLASH.

Shadow DOM est-il accessible ?

Ceci est un paragraphe dans un shadow root.

C'est parfaitement lisible, youpi. Je saute de joie.

OUI, C'EST LU

@DIOU
PENSEZ-Y...