

# The Shadow DOM

## INTRODUCTION

The Shadow DOM API provides DOM encapsulation: it hides what is not necessary!

This is already used by browsers' developers for `<audio>` or `<video>` elements, and also for the new `<input type=date>`, `<input type=color>` elements, etc.

### **The three rules of Shadow DOM:**

1. With Shadow DOM, elements are associated with a new kind of node: *a shadow root*.
2. An element that has a shadow root associated with it is called *a shadow host*.
3. *The content of a shadow host isn't rendered; the content of the shadow root is rendered instead.*

## AN EXAMPLE USING THE SHADOW DOM: THE `<VIDEO>` ELEMENT

Let's have a look at a simple `<video>` element in Google Chrome.

Open [this JSBin example](#) in your browser, and fire up the devtools console (F12 on Windows/Linux, Cmd-Alt-i on Mac OS):

Click on the "Elements" tab in the devtools, or use the magnifying glass and click on the video, to look at the the DOM view of the video element. You will see the exact HTML code that is in this example, but you cannot see the elements that compose the control bar. You don't have access to the play button, etc.

In the DOM view, we cannot see the content of this part of the video element



Elements Network Sources Timeline Profiles Resources Audits Console AngularJS

```

<video controls autoplay>
  <!-- I have three versions of the video encoded with
       different codecs. The browser will automatically
       choose the first one it knows it can play. -->
  <source src="http://html5doctor.com/demos/video-canvas-magic/video.webm"
          type="video/webm">
  <source src="http://html5doctor.com/demos/video-canvas-magic/video.ogg"
          type="video/ogg">
  <source src="http://html5doctor.com/demos/video-canvas-magic/video.mp4"
          type="video/mp4">
</video>
```

Styles Computed »

element.style { }

body { user agent stylesheet  
display: block;  
margin: 8px; }

margin 8  
border -  
padding -

html body

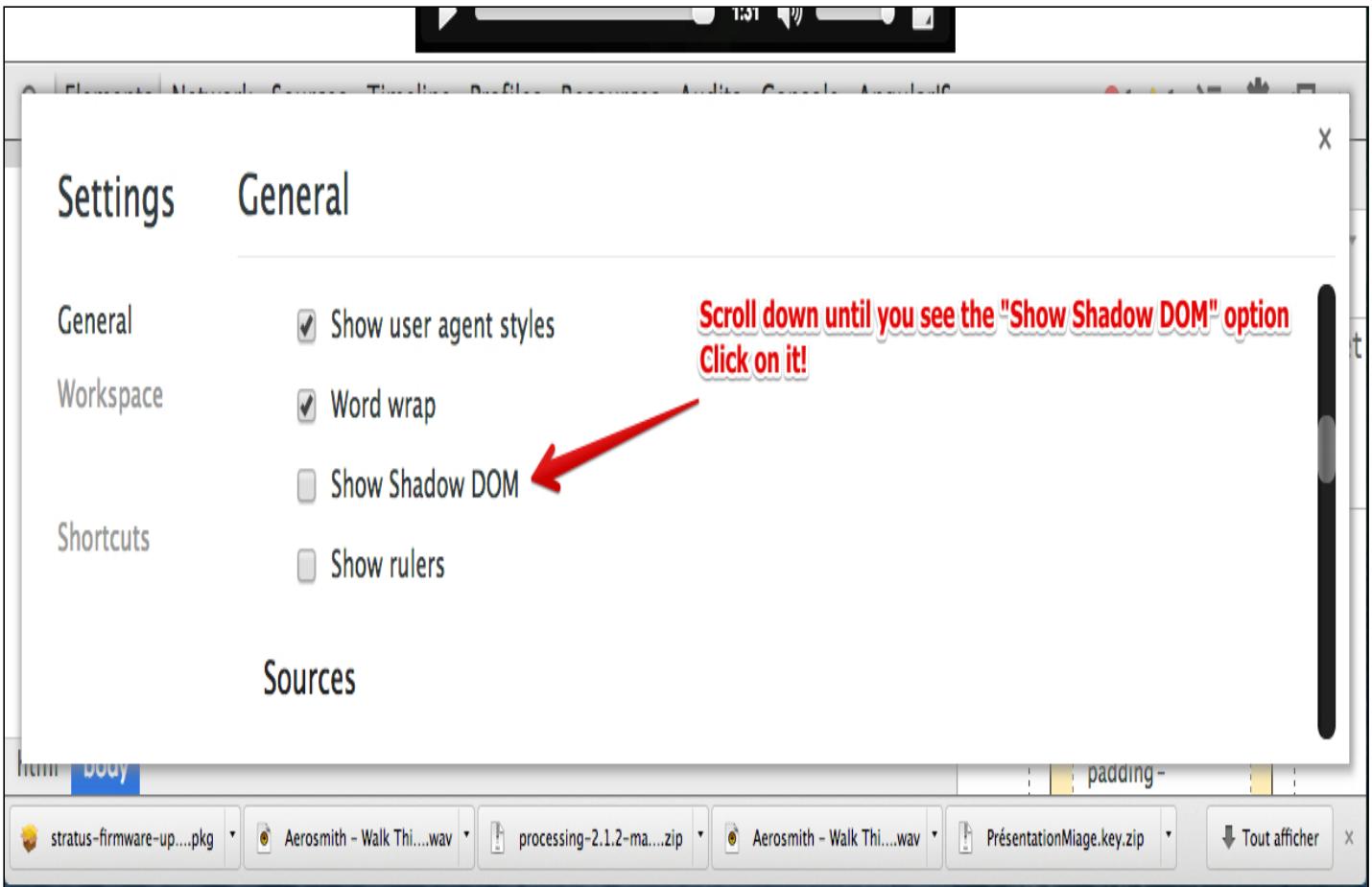
stratus-firmware-up...pkg Aerosmith - Walk Thi...wav processing-2.1.2-ma...zip Aerosmith - Walk Thi...wav PrésentationMiage.key.zip Tout afficher

Now, let's look behind the scenes, and see the Shadow DOM associated with the `<video>` element.

First, click on the small gear icon to open the option panel of the devtools:



The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. A red arrow points from the top right towards the gear icon in the toolbar. Below the toolbar, a warning message is displayed: 'Click here to show the options' over a red box, which is a standard UI element for expanding detailed information in DevTools.



Then look for the video element again within the DOM view. You should see something new:

```
<!DOCTYPE html>
<html>
  <script id="tinyhippos-injected">...</script>
  <head>...</head>
  <body>
    <video controls autoplay>
      <#shadow-root (user-agent)>
```

There is a "shadow root" associated with the <video> element!  
This element is a "shadow host", it hides a complete sub-DOM.

Styles Computed »

element.style { }

body { user agent stylesheet } display: block; margin: 8px; }

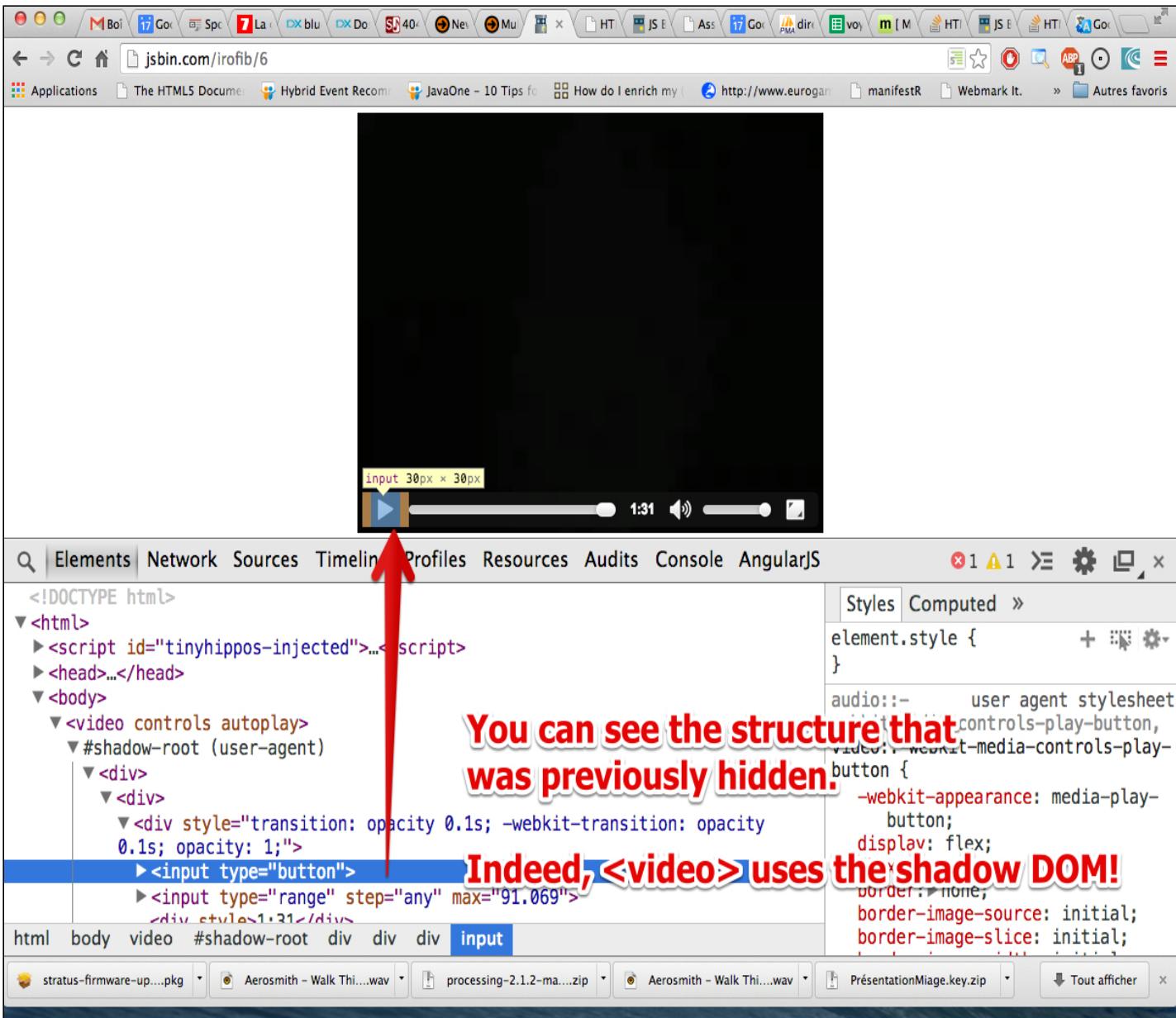
margin 8

border -

padding -

html body

Now, open this shadow root by clicking on it in the DOM view, and move the mouse pointer over the different elements:



Chrome developers are already using the shadow DOM to define their own Web Components, such as `<video>` or `<audio>` elements! And they use the Shadow DOM to hide the internal plumbing.

Furthermore, there is a kind of "boundary" around the `<video>` element, so that external CSS cannot interfere. The content of the `<video>` element is sort of *sandboxed* (protected from external CSS selectors, for example, or cannot be accessed using `document.querySelector()`, nor inspected by default, using a DOM inspector).

Browser developers have been using Web Components for a while, and now

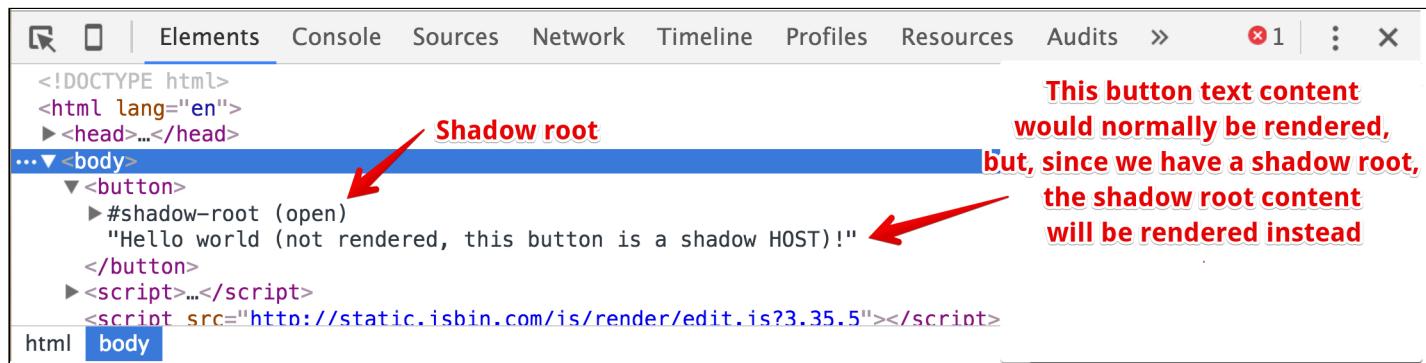
it's possible for any Web developer!

## A SIMPLE EXAMPLE OF SHADOW DOM USAGE

Let's have a look at a very simple example:

```
<button>Hello, world (not rendered)!</button>
<script>
  var host = document.querySelector('button');
  var root = host.createShadowRoot();
  root.textContent = 'the shadow root node is rendered';
</script>
```

Lines 3-5 show how to associate a shadow root to an existing element. In this example, the `<button>` defined at *line 1* is a shadow host, and it is associated with a shadow root that just contains some text (*line 5*).



This example illustrates the three rules of the shadow DOM. Let's look at them again:

### The three rules of Shadow DOM:

1. With Shadow DOM, elements are associated with a new kind of node: *a shadow root*.
2. An element that has a shadow root associated with it is called *a shadow host*.
3. *The content of a shadow host isn't rendered; the content of the shadow root is rendered instead.*

And indeed, the above example (try [the online version here at JSBin](#)) renders the content of the shadow root, not the content of the button. In the online example, try to change the text of the button (*line 1*), and you will notice that nothing changes. Then modify the text at *line 5* and look at the results!

The screenshot shows a browser window with the URL [jsbin.com/zayededa/9/edit](http://jsbin.com/zayededa/9/edit). The top bar has various tabs and icons. The main area is a code editor with tabs for HTML, CSS, JavaScript, Console, and Output. The HTML tab is selected, showing the following code:

```
<body>
<button>Hello, world (not rendered)!</button>
<script>
var host = document.querySelector('button');
var root = host.shadowRoot || host.webkitShadowRoot;
if (!root) {
  root = host.createShadowRoot ?
    host.createShadowRoot() : host.webkitCreateShadowRoot();
}

root.textContent = 'the shadow root node is rendered';
</script>
</body>
```

A red arrow points from the text "Not rendered" in the heading to the button content. Another red arrow points from the text "Rendered" in the heading to the output text "the shadow root node is rendered".

The Output panel displays the text "the shadow root node is rendered".