Using the webcam video stream

INTRODUCTION TO THE GETUSERMEDIA API

The getUserMedia API is useful for controlling a webcam video stream. This chapter presents the most interesting parts, related to the `<video>` and `<audio>` elements.

While this API is one component of the WebRTC specificationand therefore not considered a "real" part of the HTML5 specification, we still consider it relevant to the "multimedia" part of this course. The getUserMedia API, when dealing with video streams, is always used in conjunction with the `<video>`element.

TYPICAL USE OF THE GETUSERMEDIA API WITH A WEBCAM

The main idea is to set the `src` attribute of a `<video>` element to the live video stream object coming out of the webcam. To get this stream, you'll have to call the`navigator.getUserMedia(params, onSuccess, onError)` method from the getUserMedia API.

The stream is passed as a parameter to the `onSuccess()`callback, as in this typical example:

```
    <video id="myVideo" autoplay>Fallback msg here.</video>
```

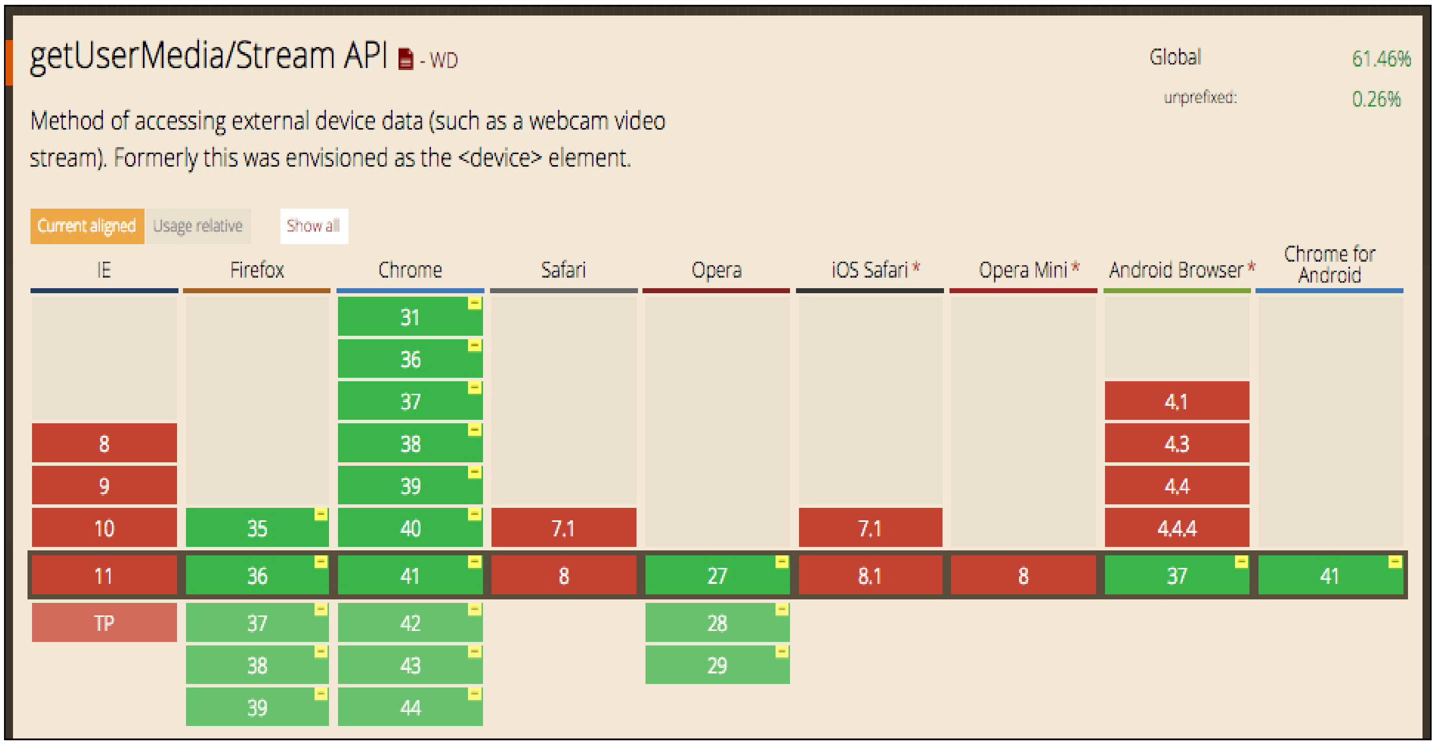
```
<script>
   function onSuccess(stream) {
     var output =document.getElementById('myVideo');
     output.src= window.URL.createObjectURL(stream);
   }
   function onError() {
     // getUserMedia API not supported, or another application is using
the webcam!
   }

   if (navigator.getUserMedia) {
     navigator.getUserMedia({video:true},onSuccess, onError);
   }
</script>
```

In fact, this example still does not work "as is". Prefixes need to be added to the API function calls, due to limitations of current browser support.

CURRENT BROWSER SUPPORT (JUNE 2015)

Since 2012, the getUserMedia API has been supported by Google Chrome, Firefox and Opera, both on desktops and mobile devices, but you still need to use the prefixed version of the API (i.e. call **webkit**`GetUserMedia` or **moz**`GetUserMedia`instead of `getuserMedia`). While Internet Explorer does not support it, it's coming soon to Microsoft Edge, IE's successor, as announced by Microsoft.



getUserMedia/Stream API - WD

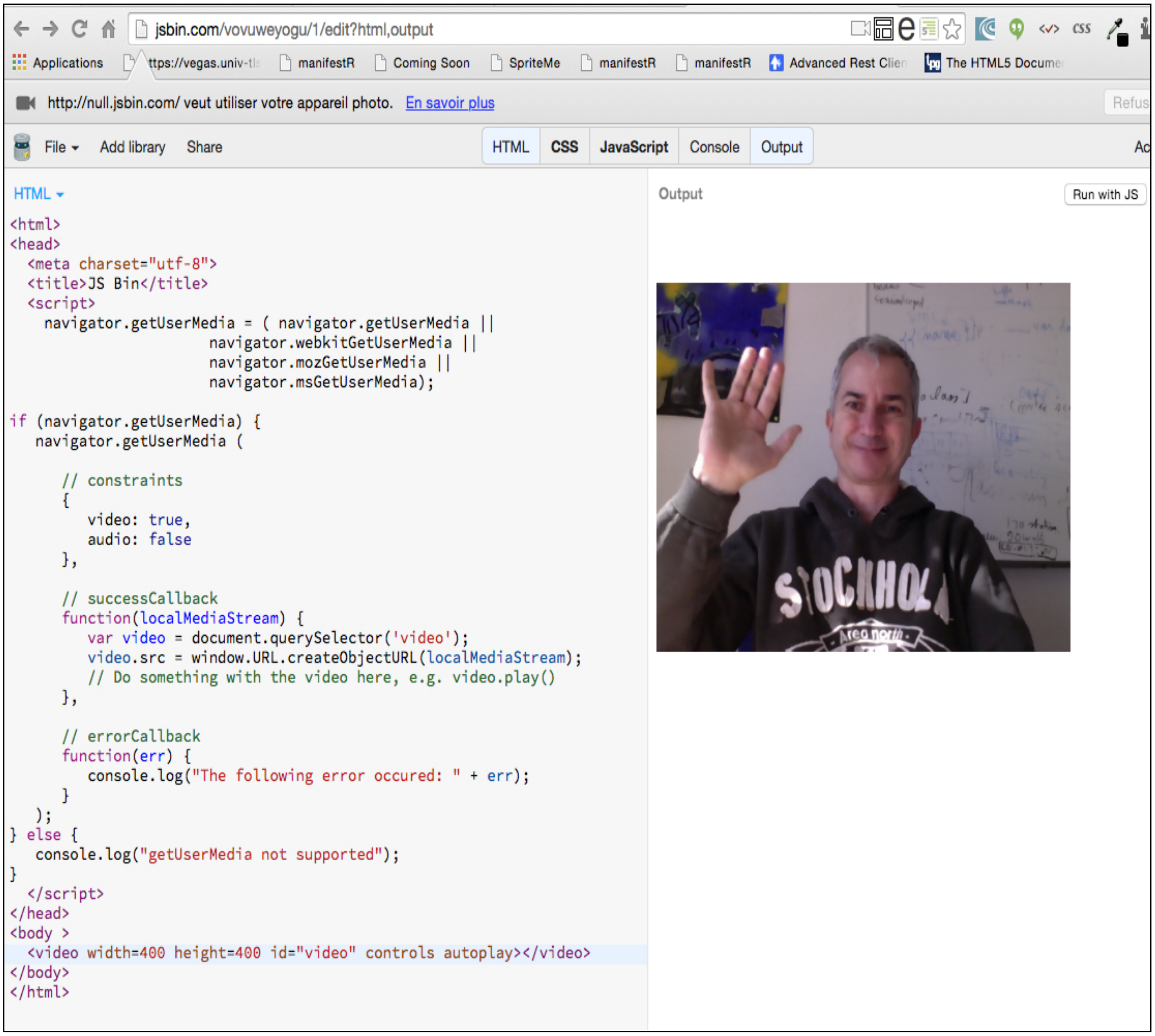
Global 61.46%
unprefixed: 0.26%

Method of accessing external device data (such as a webcam video stream). Formerly this was envisioned as the <device> element.

Current aligned | Usage relative | Show all

IE	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		31						
		36						
		37					4.1	
8		38					4.3	
9		39					4.4	
10	35	40	7.1		7.1		4.4.4	
11	36	41	8	27	8.1	8	37	41
TP	37	42		28				
	38	43		29				
	39	44						

SIMPLE WEB CAMERA DISPLAY THAT WORKS ON ALL BROWSERS WITH SUPPORT FOR THE GETUSERMEDIA API



```
<html>
<head>
  <meta charset="utf-8">
  <title>JS Bin</title>
  <script>
    navigator.getUserMedia = ( navigator.getUserMedia ||
                               navigator.webkitGetUserMedia ||
                               navigator.mozGetUserMedia ||
                               navigator.msGetUserMedia);

if (navigator.getUserMedia) {
   navigator.getUserMedia (

      // constraints
      {
         video: true,
         audio: false
      },

      // successCallback
      function(localMediaStream) {
         var video = document.querySelector('video');
         video.src = window.URL.createObjectURL(localMediaStream);
         // Do something with the video here, e.g. video.play()
      },

      // errorCallback
      function(err) {
         console.log("The following error occured: " + err);
      }
   );
} else {
   console.log("getUserMedia not supported");
}
  </script>
</head>
<body >
  <video width=400 height=400 id="video" controls autoplay></video>
</body>
</html>
```

This example is available online: JS Bin example

It's similar to the first example above except that we use the prefixed versions of `getUserMedia(...)`. This example works on Opera/Chrome/Firefox on Desktop and Mobile.

Another noticeable difference is the haircut of your instructor, but that's another story :-)

Source code:

```
<html>
<head>
 <meta charset="utf-8">
 <title>JS Bin</title>
 <script>
    navigator.getUserMedia = (navigator.getUserMedia ||
                              navigator.webkitGetUserMedia ||
                              navigator.mozGetUserMedia ||
                              navigator.msGetUserMedia);

    if (navigator.getUserMedia) {
        navigator.getUserMedia (
            // constraints
            {
                video: true,
                audio: false
            },

            // successCallback
            function(localMediaStream) {
                var video =document.querySelector('video');
                video.src =window.URL.createObjectURL(localMediaStream);
            },

            // errorCallback
            function(err) {
                console.log("The following error occured: " + err);
            }
        );
    } else {
        console.log("getUserMedia not supported");
    }
 </script>
</head>
<body >
     <video width=200 height=200id="video" controls autoplay></video>
</body>
</html>
```

KNOWLEDGE CHECK 2.5.1 (NOT GRADED)

What is getUserMedia?

- ○ A JavaScript API that can be used to redirect the webcam video stream to a video element
- ○ An upcoming API that is not available yet on browsers, but can be emulated by the video element
- ○ An API which only works with WebRTC for audio conferencing

CHECK
