

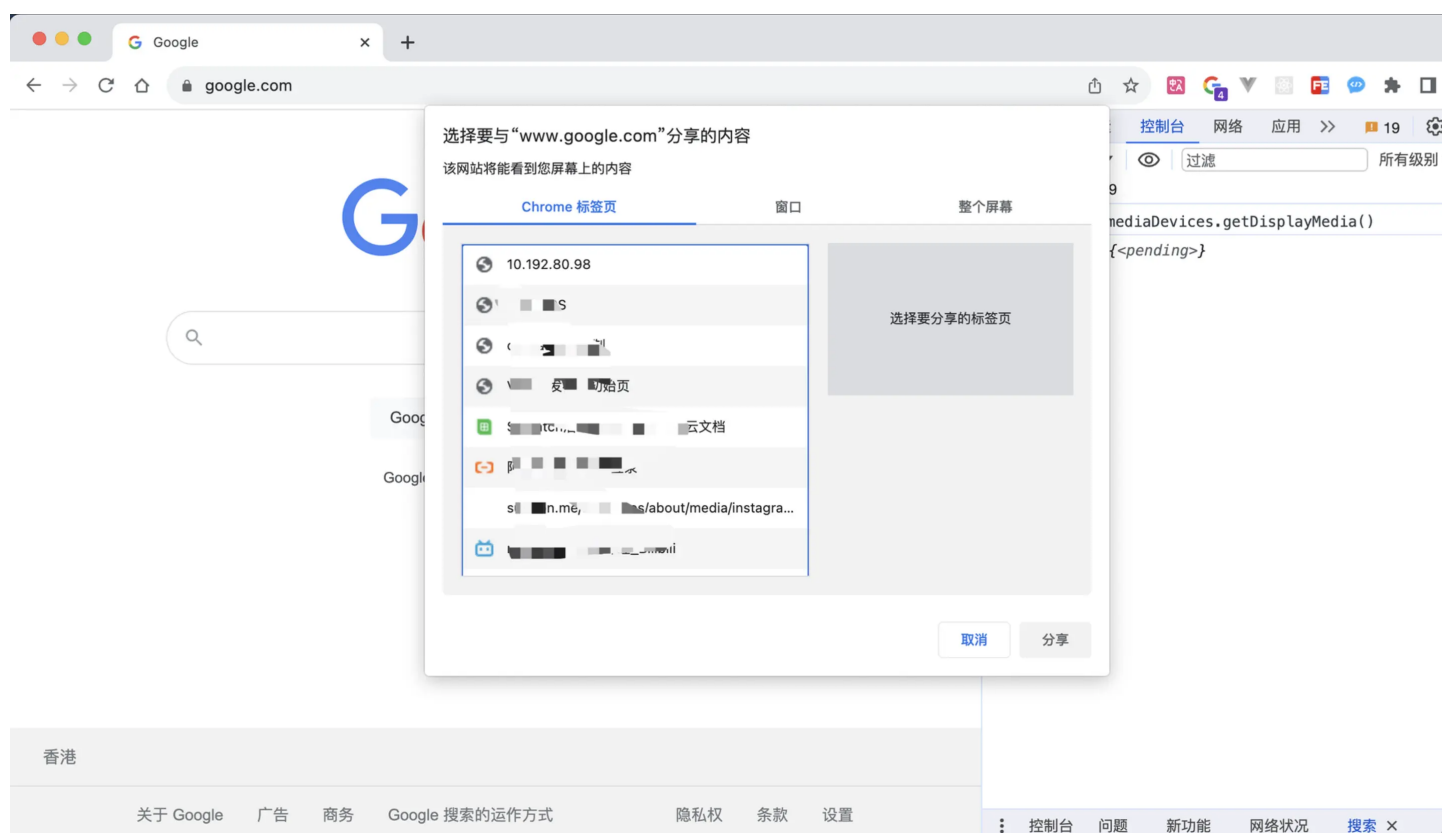
# 纯前端也可以实现「用户无感知录屏」？

## 前言

要在 JavaScript 中实现屏幕录制，可以使用

`navigator.mediaDevices.getDisplayMedia()` 方法来获取屏幕的媒体流。然后，可使用 `MediaRecorder` 对象将媒体流录制为视频文件。

但该方法会在浏览器弹出一个授权窗口，让用户选择要分享的内容，这不可实现“无感知”。



如果真正做到无感知，那我们就不能借助浏览器或者系统系统的能力了。我们能做的就只能是通过js去操作了。

要在页面内直接录制视频似乎并不容易，没有现成的开源库可以使用，也没有很好的想法。

那我们换一个思路，视频是由帧组成的，我们是否可以不断的截图，然后组合成一段视频？好像是可以的。



## 效果



图片 加载失败

## 页面

先写一个简单的页面：

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <title>Canvas视频录制</title>
7     <link rel="stylesheet" href="styles.css">
8 </head>
9
10 <body>
11     <main>
12         <div class="buttons">
13             <button class="start-btn">开始录制</button>
14             <button class="pause-btn">暂停录制</button>
15             <button class="resume-btn">继续录制</button>
16             <button class="stop-btn">结束录制</button>
17         </div>
18         <div id="box">
19             <section class="content">
20                 <h2>TODO LIST</h2>
```

```

21         <div class="background-div">
22             <button class="background-btn">切换背景颜色</button>
23         </div>
24         <div id="todo-form">
25             <input type="text" class="input-field" placeholder="输入待办
事项">
26             <button type="submit" class="submit-btn">提交</button>
27         </div>
28         <div class="list"></div>
29     </section>
30 </div>
31 <img src="" alt="" class="hidden">
32 </main>
33
34 <script src="https://cdn.bootcss.com/html2canvas/0.5.0-
beta4/html2canvas.min.js" defer></script>
35 <script src="canvas.js" defer></script>
36 </body>
37
38 </html>
39

```

## 截图

实现网页的截图操作，最常用的库是 `html2canvas` 用，它可以将网页中的 HTML 元素转换为 Canvas 元素，并将其导出为图像文件。在浏览器中捕获整个页面或特定区域的截图，包括 CSS 样式和渲染效果。

```

1 const canvasFunction = () => {
2     html2canvas(box).then(canvas => {
3         const imgStr = canvas.toDataURL("image/png");
4         img.src = imgStr;
5         img.onload = function () {
6             ctx.drawImage(img, 0, 0, w, h);
7         }
8     });
9 }

```

## 合成视频

这里我们要使用到一个 API `MediaRecorder`，用于在浏览器中进行音频和视频的录制。它提供了一种简单的方式来捕获来自麦克风、摄像头或屏幕的媒体数据，并将其保存为文件或进行实时流传输。

它有以下几个常用的方法：

- `isTypeSupported()` 返回一个 Boolean 值，来表示设置的 MIME type 是否被当前用户的设备支持。
- `start()` 开始录制媒体，这个方法调用时可以通过给 `timeslice` 参数设置一个毫秒值，如果设置这个毫秒值，那么录制的媒体会按照你设置的值进行分割成一个个单独的区块，而不是以默认的方式录制一个非常大的整块内容。
- `pause()` 暂停媒体录制。
- `resume()` 继续录制之前被暂停的录制动作。
- `stop()` 停止录制。同时触发 `dataavailable` 事件，返回一个存储 `Blob` 内容的录制数据。之后不再记录。

首先创建一个 `canvas` 元素，用来保存 `html2canvas` 的截图，然后通过 `captureStream` 方法实时截取媒体流。

```
1 const w = boxBoundingClientRect.width;
2 const h = boxBoundingClientRect.height;
3 const canvas = document.createElement('canvas');
4 canvas.setAttribute('id', 'canvas');
5 canvas.setAttribute('width', w);
6 canvas.setAttribute('height', h);
7 canvas.style.display = 'none';
8 box.appendChild(canvas);
9
10 const img = document.querySelector('img');
11 const ctx = canvas.getContext("2d");
12 const allChunks = [];
13 const stream = canvas.captureStream(60); // 60 FPS recording 1秒60帧
```

通过 `canvas` 的流来创建一个 `MediaRecorder` 实例，并在 `ondataavailable` 事件中保存视频信息：

```
1 const recorder = new MediaRecorder(stream, {
2   mimeType: 'video/webm;codecs=vp9'
3 });
4
5 recorder.ondataavailable = (e) => {
6   allChunks.push(e.data);
7 };
```

最后，在停止录制时将帧信息创建 blob 并插入到页面上：

```
1 recorder.stop();
2 const fullBlob = new Blob(allChunks);
3 const videoUrl = window.URL.createObjectURL(fullBlob);
4
5 const video = document.createElement('video');
6 video.controls = true;
7 video.src = videoUrl;
8 video.muted = true;
9 video.autoplay = true;
10 document.body.appendChild(video);
```

或者可以将视频下载

```
1 recorder.stop();
2 const fullBlob = new Blob(allChunks);
3 const videoUrl = window.URL.createObjectURL(fullBlob);
4
5 let link = document.createElement('a');
6 link.style.display = 'none';
7 let fullBlob = new Blob(allChunks);
8 let downloadUrl = window.URL.createObjectURL(fullBlob);
9 link.href = downloadUrl;
10 link.download = 'canvas-video.mp4';
11 document.body.appendChild(link);
12 link.click();
13 link.remove();
```

这里，为了节省资源，只在点击按钮、输入等事件发生时才调用 `html2canvas` 截图 DOM。

如果实时记录屏也可以使用 `requestAnimationFrame` 。

## 最后

虽然实现了无感知录制屏幕，但也仅限于网页内，没有办法录制网页以外的部分。

以上的 demo 中只实现了 DOM 的录制，如果想要录制鼠标轨迹，可以增加一个跟随鼠标的元素～