**js中的audio声音播放及浏览器播放报错处理**

 本文介绍了在Web端中使用audio标签和HTML5SpeechSynthesisUtteranceAPI播放声音的两种方式，强调了在调用audio.play()时需确保用户与页面交互以避免DOMException。同时提供了处理浏览器自动播放限制的示例。

摘要由CSDN通过智能技术生成

在web端中常常会用到播放声音用来提示消息、告警等等。

在这有两种实现方式：

采用audio调取声音播放url  
采用html5新增的API(SpeechSynthesisUtterance：用于将指定文字合成为对应的语音.也包含一些[配置项](https://so.csdn.net/so/search?q=%E9%85%8D%E7%BD%AE%E9%A1%B9&spm=1001.2101.3001.7020),指定如何去阅读(语言,音量,音调)等）  
1.采用audio调取声音播放url（canplaythrough事件为音频加载完成且无卡顿和缓冲时触发）

1. var url = */\* 音频播放地址 \*/*
2. var audio = new Audio();
3. audio.src = url;
4. audio.addEventListener("canplaythrough", event => {
5. */\* 音频可以播放；如果权限允许则播放 \*/*
6. audio.play();
7. });

2 .采用SpeechSynthesisUtterance

1. let utterThis = new SpeechSynthesisUtterance('干饭啦！');
2. speechSynthesis.speak(utterThis);

此外还有其他的配置项

speech.lang 获取并设置话语的语言（en-US、zh-CN）  
speech.pitch 获取并设置话语的音调 (值越大越尖锐，range:0-2, default:1, float)  
speech.rate 获取并设置说话的速度 (值越大语速越快, range:0.1-10, default:1, float)  
speech.text 获取并设置说话时的文本  
speech.voice 获取并设置说话的声音  
speech.volume 获取并设置说话的音量 (range: 0-1, default:1, float)  
speech.onboundary  
speech.onend 播放结束的回调  
speech.onerror 播放出现错误的回调  
speech.onmark 当读到标记文本时的回调  
speech.onpause 播放暂停  
speech.onresume 播放重启  
speech.onstart 播放开始的回调  
案例：

1. function speak (msg) {
2. var utterThis = new SpeechSynthesisUtterance()
3. utterThis.text = msg
4. utterThis.pitch = 1 *// 获取并设置话语的音调(0-2 默认1，值越大越尖锐,越低越低沉)*
5. utterThis.rate = 0.9 *// 获取并设置说话的速度(0.1-10 默认1，值越大语速越快,越小语速越慢)*
6. utterThis.volume = 100 *// 获取并设置说话的音量*
7. utterThis.lang = 'zh-CN' *// 设置播放语言*
8. *// 语音播报结束时的回调*
9. utterThis.onend = (event) => {
10. *// 语音播报完需要清空展示内容*
11. }
12. }
13. speak('干饭啦！')

 但是在浏览器常常会报错

DOMException: play() failed because the user didn't interact with the document first。意思就是调用audio.play()方法之前，用户必须先和页面有交互，否则调用失败。

这里的交互是指必须用户与页面发生了事件，单击 [onclick](https://so.csdn.net/so/search?q=onclick&spm=1001.2101.3001.7020)，双击 ondblclick，按下 [onmousedown](https://so.csdn.net/so/search?q=onmousedown&spm=1001.2101.3001.7020)，松开 onmouseup，右击 oncontextmenu ，按下 onkeydown，松开 onkeyup 等

1.可以在用户进页面上展示一个欢迎框，点确定调取click事件之后就可以播放了

2.就是当有声音进来采用try....catch方法弹出框提示用户开启语音播放

1. var url = "" *// 播放地址;*
2. audio = new Audio();
3. audio.src = url;
4. audio.addEventListener("canplaythrough", event => {
5. *// 如果视频播放失败在这里弹出提示框*
6. */\* 音频可以播放；如果权限允许则播放 \*/*
7. audio.play().then((res) => {
8. console.log('视频播放成功')
9. })
10. .catch((error) => {
11. layer.open({
12. title: '提示',
13. shade: 0.6, *//遮罩透明度*
14. content: '由于您的浏览器设置，声音无法自动播放，请点击确认按钮授权播放。',
15. btn: '确认',
16. btnAlign: 'c', *//按钮居中*
17. yes: function (index) {
18. layer.close(index);
19. audio.play()
20. },
21. cancel: function (index) {
22. layer.close(index);
23. audio.play()
24. },
25. });
26. })
27. });