

Web Audio Tutorial

9/4, 2015

Your Goal

- Learn what digital audio is and how it works
- Implement the concept of digital audio on the Web

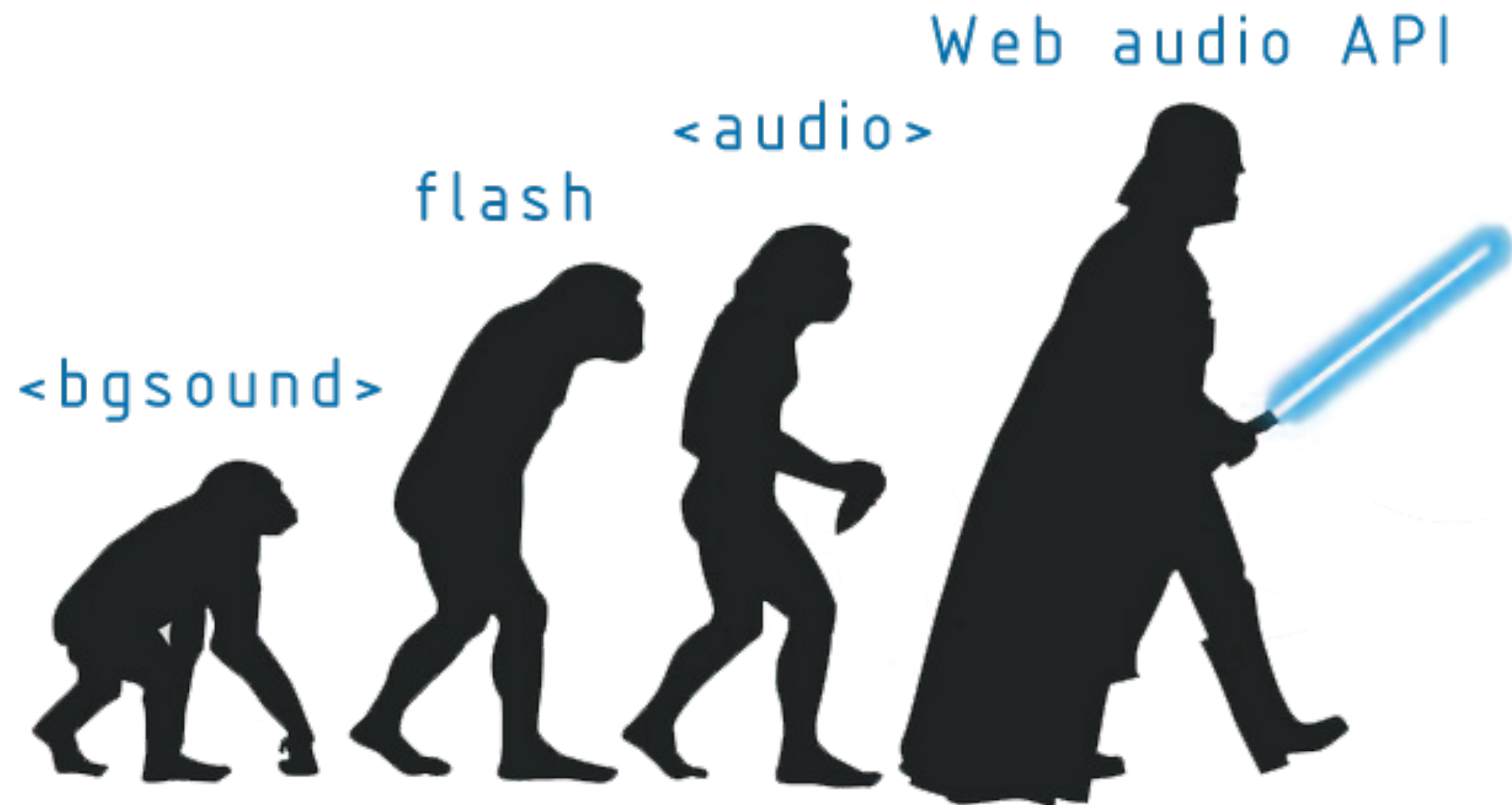
My Goal

- Introduce you Web Audio API (+ basic Web programming)
- Provide some starting points and references

Why Web?

- Platform independent
- Network-familiar
- Easy distribution

History of Web Audio



The Web

“The World Wide Web (www, W3) is an **information space** where documents and other web resources are identified by **URIs**, interlinked by hypertext links, and can be accessed **via the Internet**.”

The Web You See

KAIST Portal System(2) x

← → ↺ https://portal.kaist.ac.kr/ennotice/today_notice ☆ 📱 ☰

English | My 설정 | 로그아웃 | 📄 사이트맵 | 📱 Mobile

KAIST Portal

🔍 동아리 홍보물 KAMF WELCOME KAIST 등록금 미휴 <>

📱 다양한 소식을 Por **PUSH 알림**으로

공지사항 | **게시판** | **캘린더** | **고객만족** | **PUSH 알림** | **포탈안내** | **바로가**

공지사항

뉴스레터

업무공지사항

International Community

IT 서비스

체육/건강관리/클리닉

연구개발사업 공지사항

업무편람/양식

세미나/행사공지

학생공지사항

수강/학적/논문

리더십/인턴/상담

생활관/장학/복지

교과과정신설변경

취업

의총(총학고지)

오늘의 공지사항

 오늘의 공지사항 게시판입니다.

PUSH 알림 구독을 설정하시면 해당 게시판에 글 등록시 포탈 앱으로 푸시 알림 메시지를 받아 보실 수 있습니다.
공지사항은 직원 그룹만 게시 할 수 있습니다.
다른 그룹의 구성원께서는 관련 담당 직원에게 문의하시기 바랍니다.

검색분류

Total **24651**

제목	소속	작성자	조회수
<모집 마감> 시설팀 근로장학생 모집(2015학년도 가을학기) 📄	시설팀	오승현	332
[ISSS] Office closed on August 27th~28th	국제교원 및 학생지원팀	정효진	28
가동아리 국제화 인증제 선발안내(상금, 상패수여) 📄	국제교원 및 학생지원팀	임지윤	70
중앙도서관(학술정보개발팀)에서 근로장학생을 모집합니다.	학술정보개발팀	이기범	257
[채용] KAIST CT연구소 / Augmented Human 연구센터 전임...	문화기술연구소	양현	80

Web Page

- Web Page : HTML + CSS + Javascript (+...)

HTML

- Hypertext Markup Language
- Hypertext: text that uses hyperlink
- Markup Language: a language that uses `<bold>mark</bold>`
- Usually, HTML provides the structure and contents of a page

HTML Tutorial

Edit This Code:

See Result »

Result:

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
<h1>This is a Heading</h1>
<p>This is a paragraph.</p>
<div>
  This is a division.
</div>
</body>

</html>
```

This is a Heading

This is a paragraph.

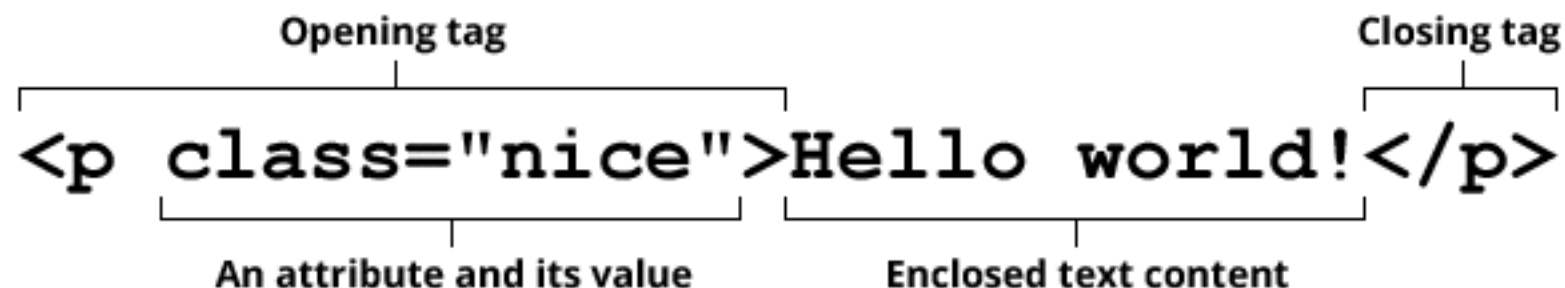
This is a division.

HTML Elements

An element is a part of a webpage.
In XML and HTML, an element may contain a data item or a chunk of text or an image, or perhaps nothing.

A typical element includes an opening tag, attributes, content, and a closing tag:

Anatomy of an HTML element



Consult <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

<https://developer.mozilla.org/en-US/docs/Glossary/Element>

Style

- Specify how a tag or an HTML element will look like
- Often separated as a single file called CSS cascade

Edit This Code:See Result »Result:

```
<!DOCTYPE html>
<html>

<head>
  <title>Page Title</title>
</head>

<body>
<h1 style="font-size:20px">This
is a Heading</h1>
<p style="color:red">This is a
paragraph.</p>
<div style="border-style:solid">
  This is a division.
</div>
</body>

</html>
```

This is a Heading

This is a paragraph.

This is a division.

CSS-Striped Web Page



오늘의 공지사항



오늘의 공지사항 게시판입니다.

PUSH 알림 구독을 설정하시면 해당 게시판에 글 등록시 포탈 앱으로 푸시 알림 메시지를 받아 보실 수 있습니다.

공지사항은 직원 그룹만 게시 할 수 있습니다.

다른 그룹의 구성원께서는 관련 담당 직원에게 문의하시기 바랍니다.

검색

테이블명

검색분류 제목 검색어 검색

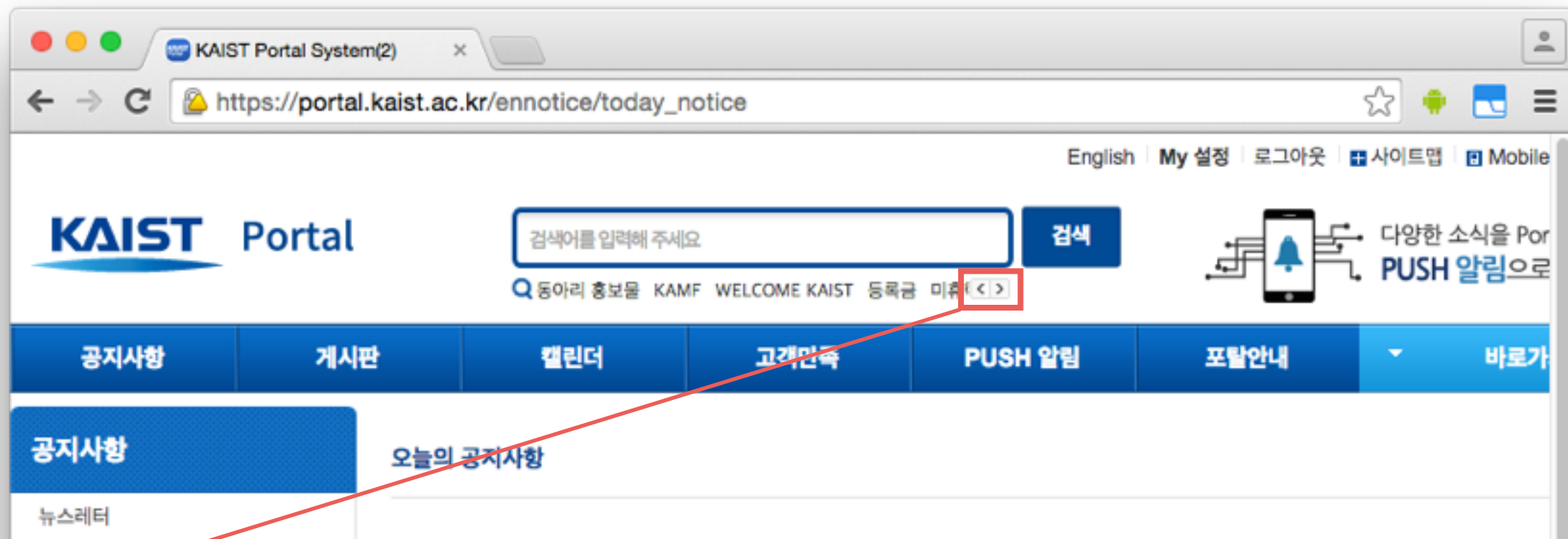
• Total 24651

제목	테이블명	소속	작성자	조회수	작성일
<모집 마감> 시설팀 근로장학생 모집(2015학년도 가을학기)	시설팀		오승현	332	2015.08.25
[ISSS] Office closed on August 27th~28th	국제교원 및 학생지원팀		정효진	28	2015.08.25
가동아리 국제화 인증제 선발안내(상급, 상패수여)	국제교원 및 학생지원팀		임지윤	70	2015.08.25
중앙도서관(학술정보개발팀)에서 근로장학생을 모집합니다.	학술정보개발팀		이기범	257	2015.08.25
[채용] KAIST CT연구소 / Augmented Human 연구센터 전임연구원 모집(연구교수, 포닥, 병역특례 등)	문화기술연구소		양현	80	2015.08.25

Do we call this programming?

Javascript

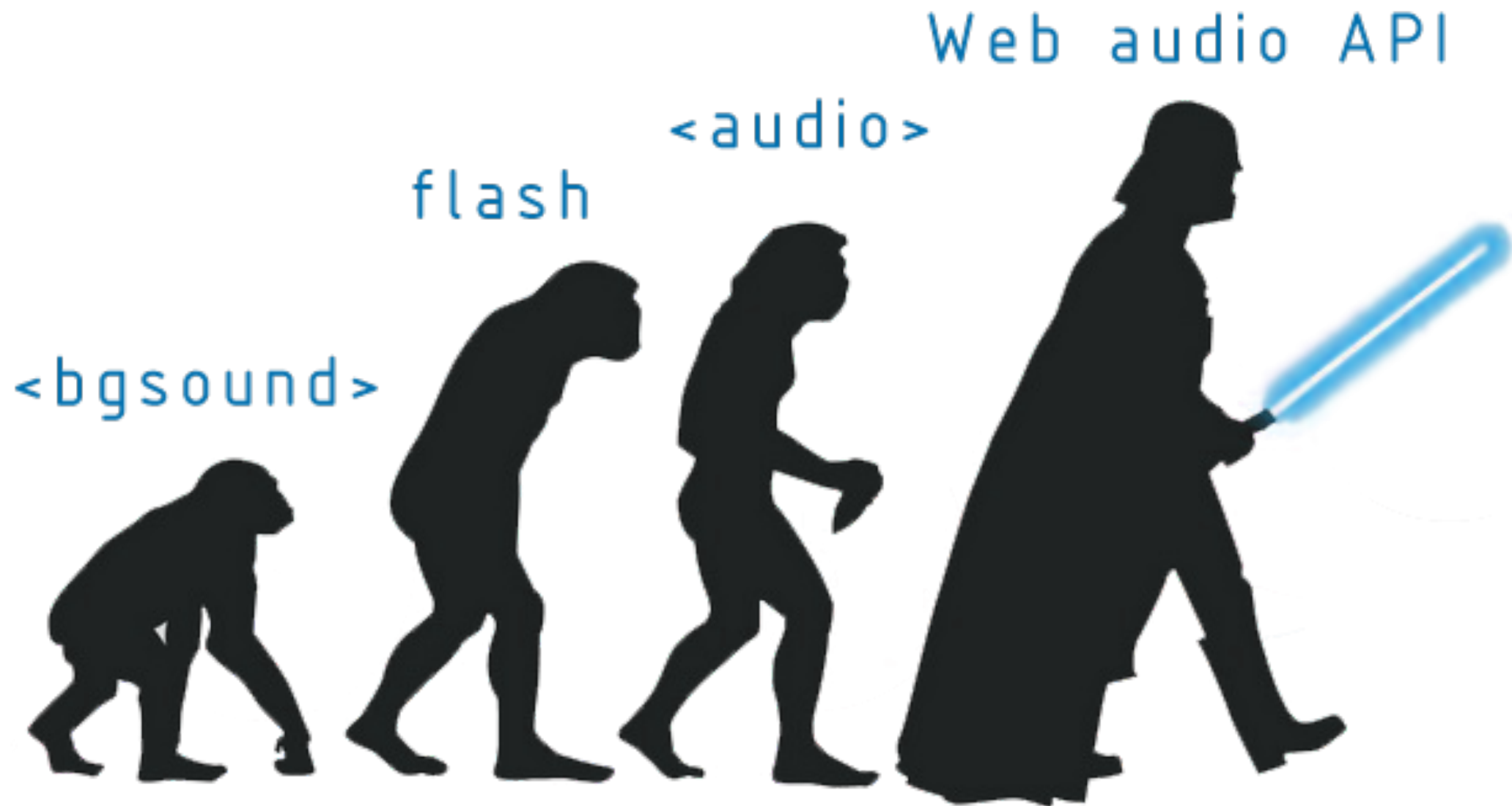
- A programming language that is (often) used for Dynamic Web pages
- Consult <https://developer.mozilla.org/en-US/docs/Web/JavaScript>



```
<a href="javascript:void(0);"  
onclick="fn_trendKeyAnimation('R');"  
style="position: absolute; right: 0px;">  
    
</a>
```

```
function fn_trendKeyAnimation(direction) {  
    var width = 0;  
    if (direction == "R") {  
        width = $('#trendKey li:last-child').width();  
        $("#trendKey").animate({  
            left: + width  
        }, 400, function () {  
            $('#trendKey li:last-child').prependTo('#trendKey');  
            $('#trendKey').css('left', '');  
        });  
    } else { /* hidden for compactness */  
    }  
}
```

Back To The History



The History

- `<bgsound>`: an element that can play a midi file
- `flash`: yeah, the Flash you know
- `<audio>`: an element that can play audio files

<audio>

Edit This Code:

See Result »

Result:

```
<!DOCTYPE html>
<html>
<body>

<audio controls>
  <source src="horse.mp3"
type="audio/mpeg">
</audio>

</body>
</html>
```

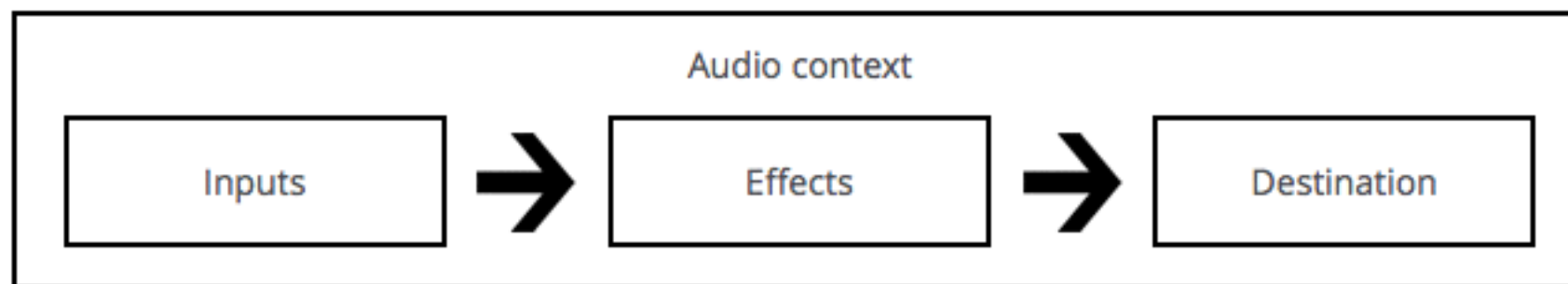


<audio>

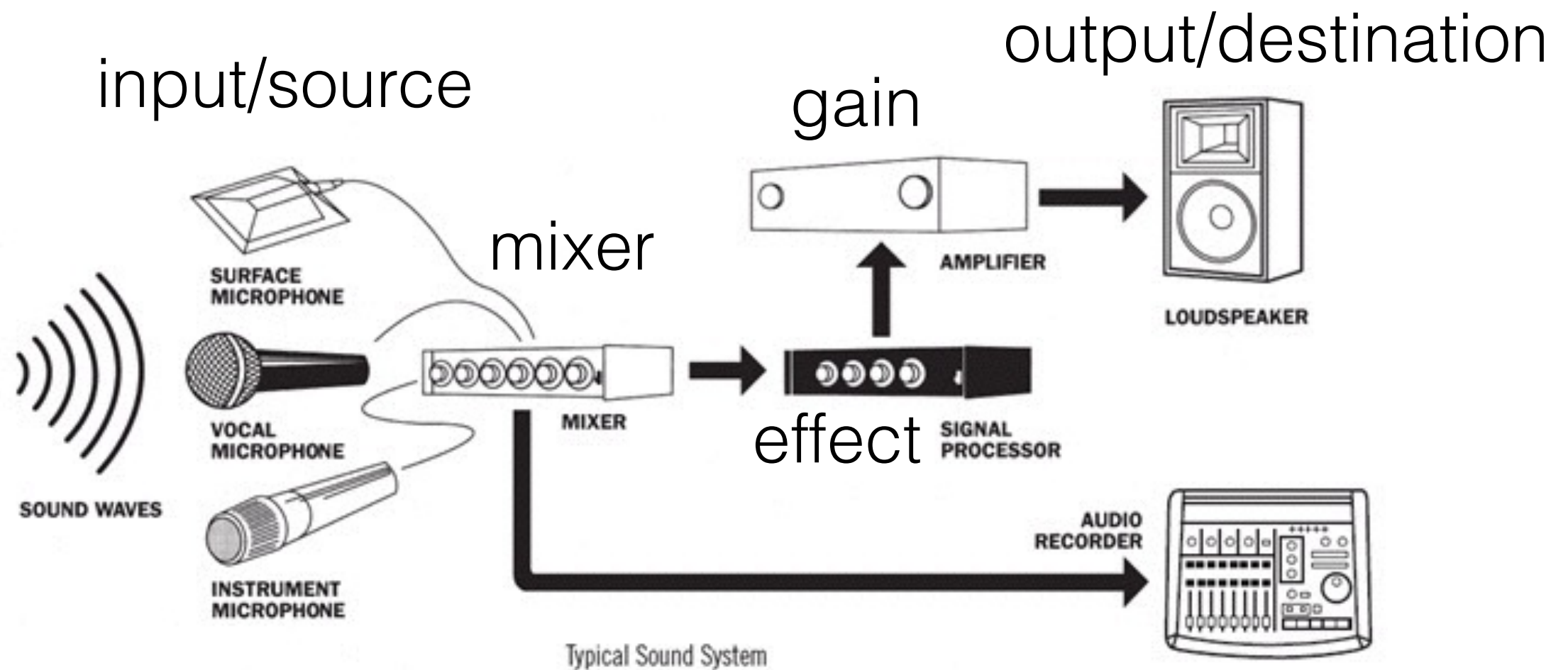
- + You can play, stop, seek, or even detect some events
- + Programmatically controllable (via Javascript)
- Nothing more than playing audio files

Web Audio API

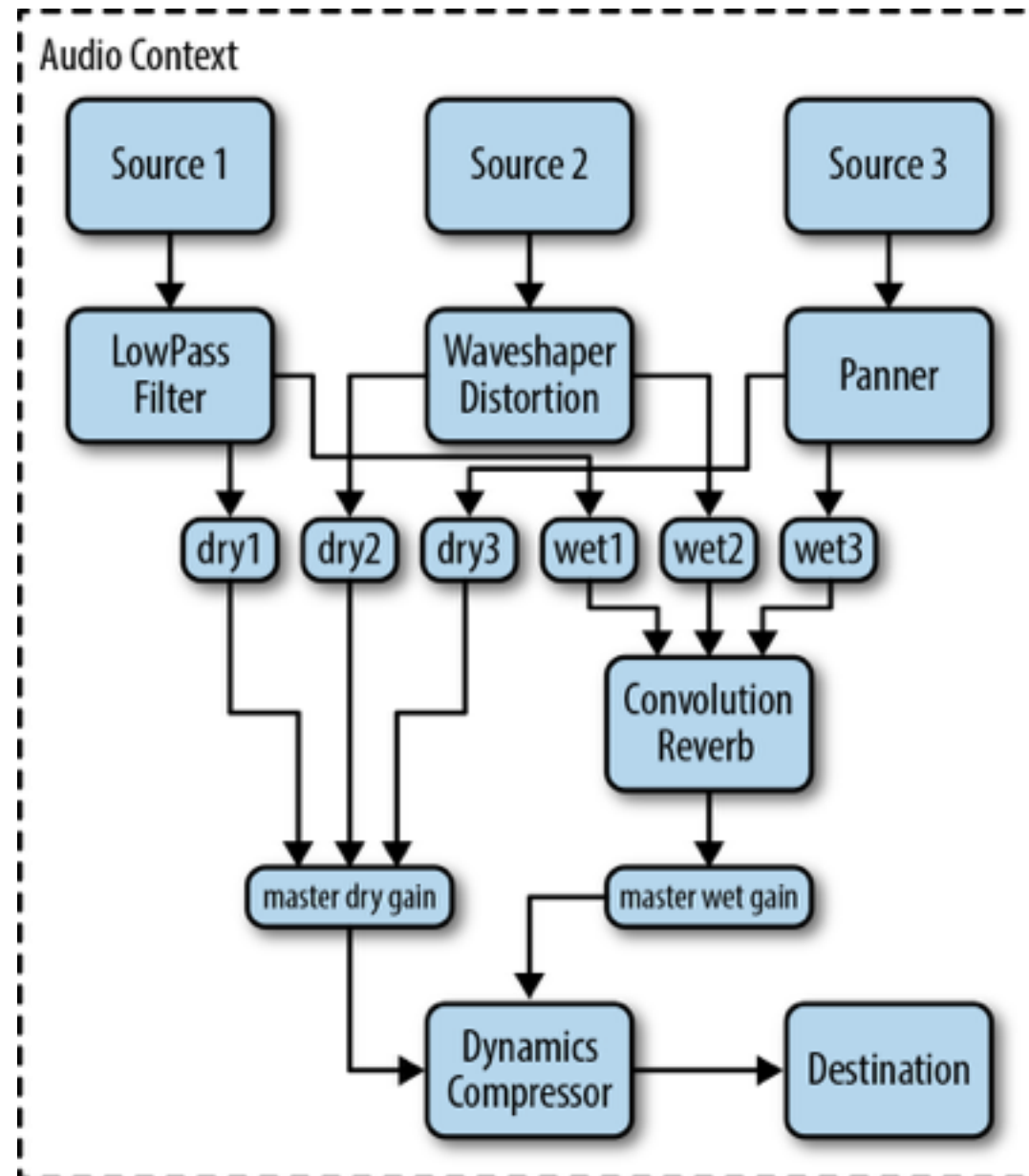
- Audio mixing, routing, signal processing, synthesis, and a bit of analysis
- Consists of an audio context and audio nodes



Real Audio System



In Web Audio



Web Audio in Action

- <https://www.soundtrap.com/studio/>
- <http://dinahmoelabs.com/plink>
- <http://webaudiodemos.appspot.com/>
- <https://chromium.googlecode.com/svn/trunk/samples/audio/samples.html>

Web Audio in Action

- <http://www.google.com/doodles/robert-moogs-78th-birthday>
- <http://onlinesequencer.net/>
- <http://www.hongkiat.com/blog/virtual-instrument-web-browser/>
- <http://patterns sketch.com/>
- <http://webaudioapi.com/samples/>

Simple Code

- Simple audio player
- Button interaction and event listening
- Audio context
- Loading and decoding audio files
- Buffer, BufferSourceNode and connecting nodes

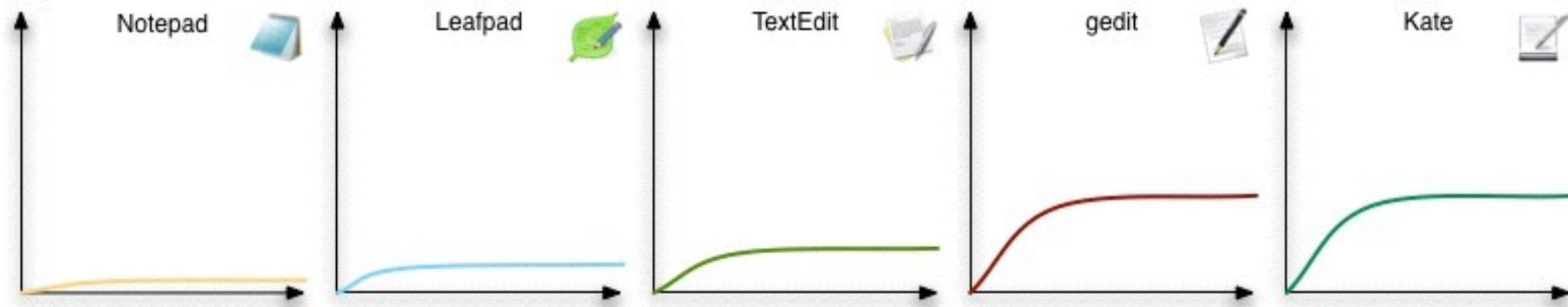
First of all, you need
a **working** web browser



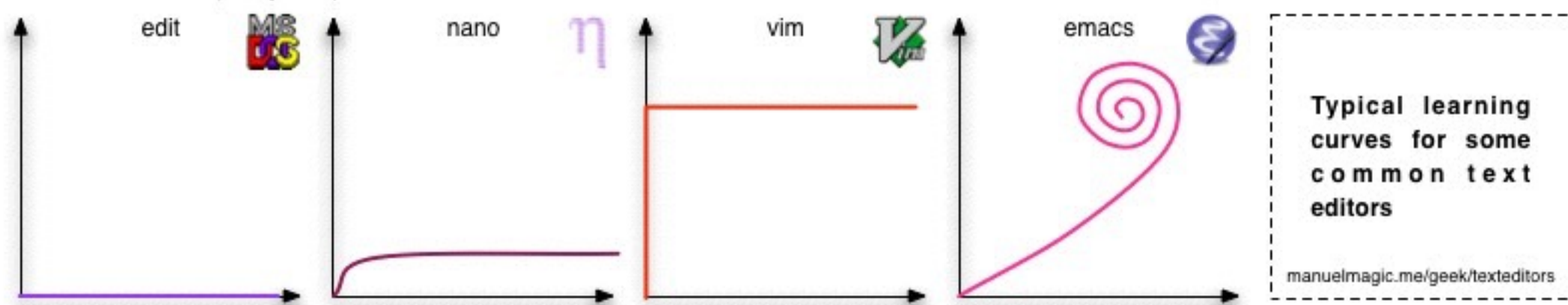
Download Chrome or Firefox (preferably Developer Edition)

...and an text editor

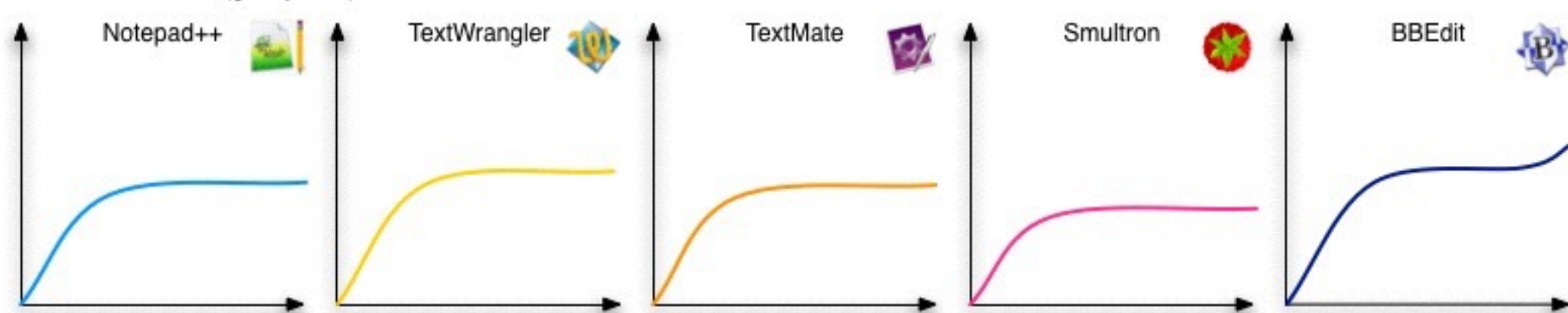
System default editors



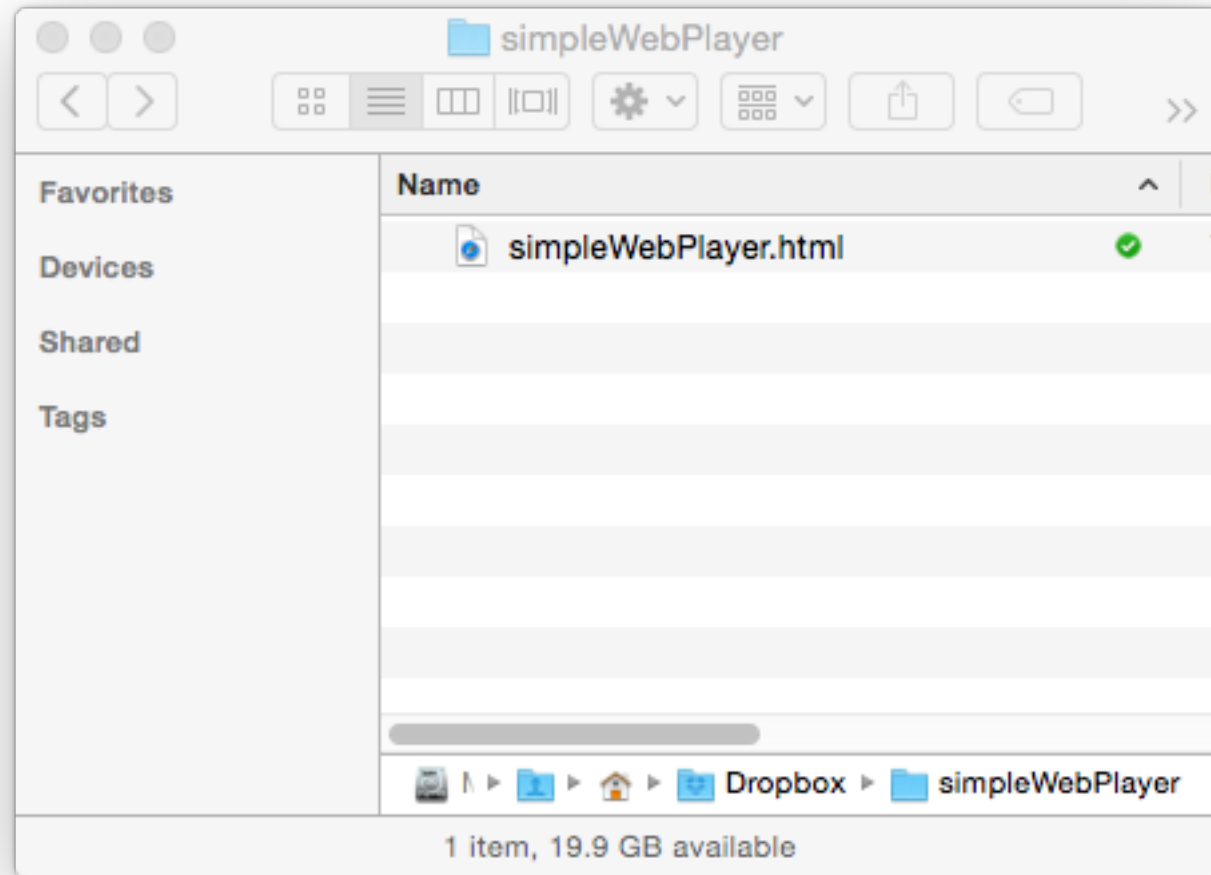
Terminal editors (nerdy stuff)



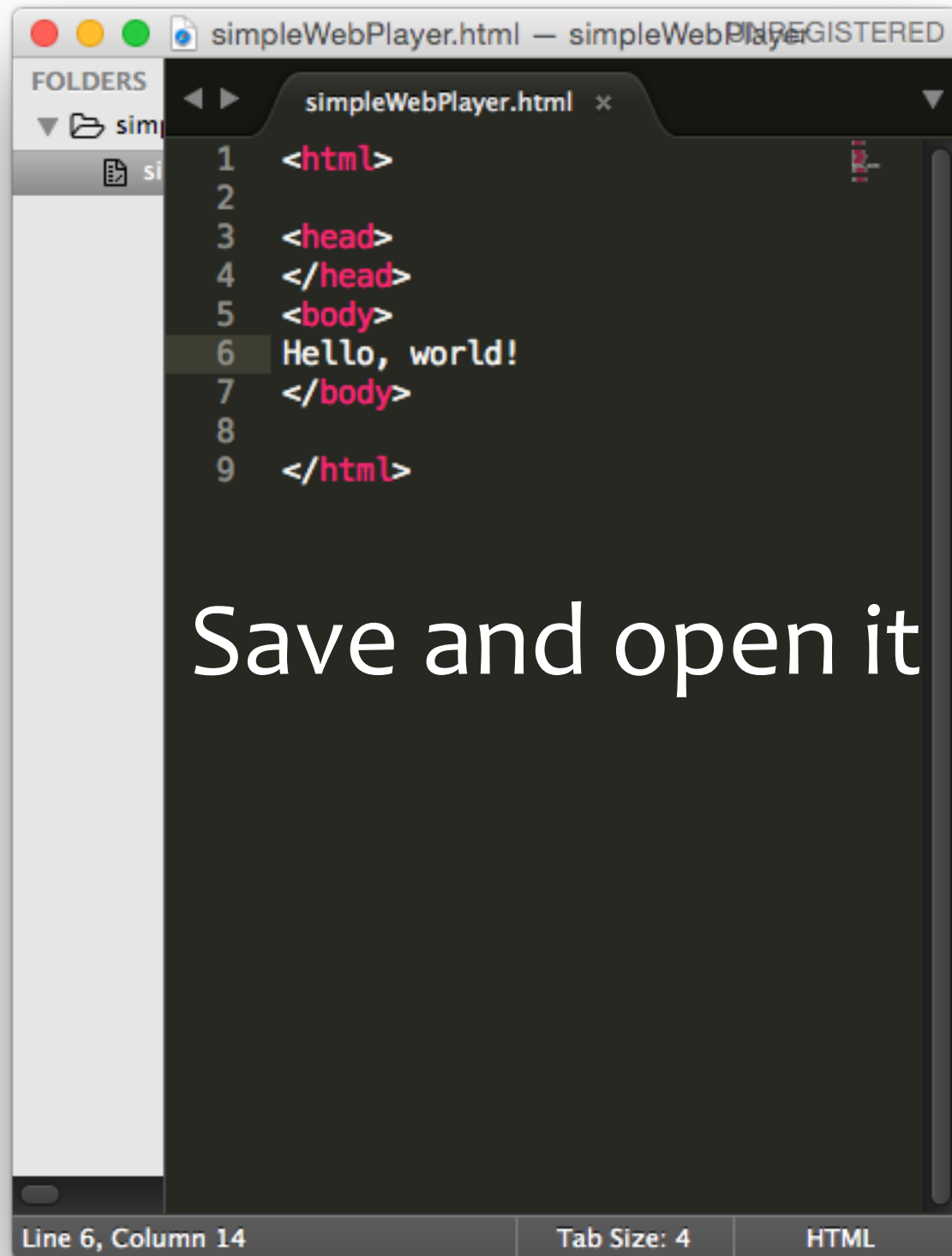
Freeware editors (geeky stuff)



Make an HTML file



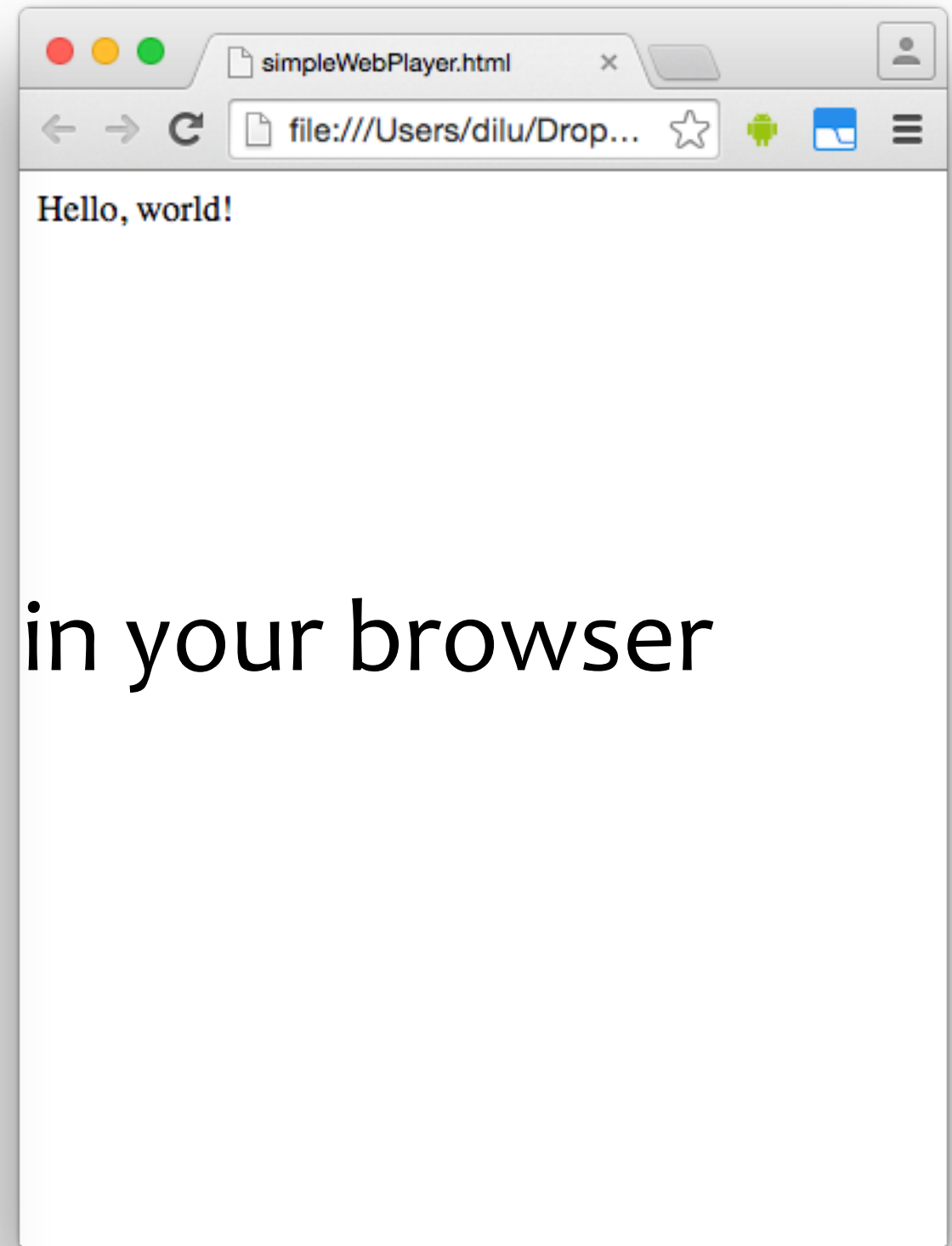
The very skeleton



A screenshot of a code editor window titled "simpleWebPlayer.html — simpleWebPlayer". The editor shows the following HTML code:

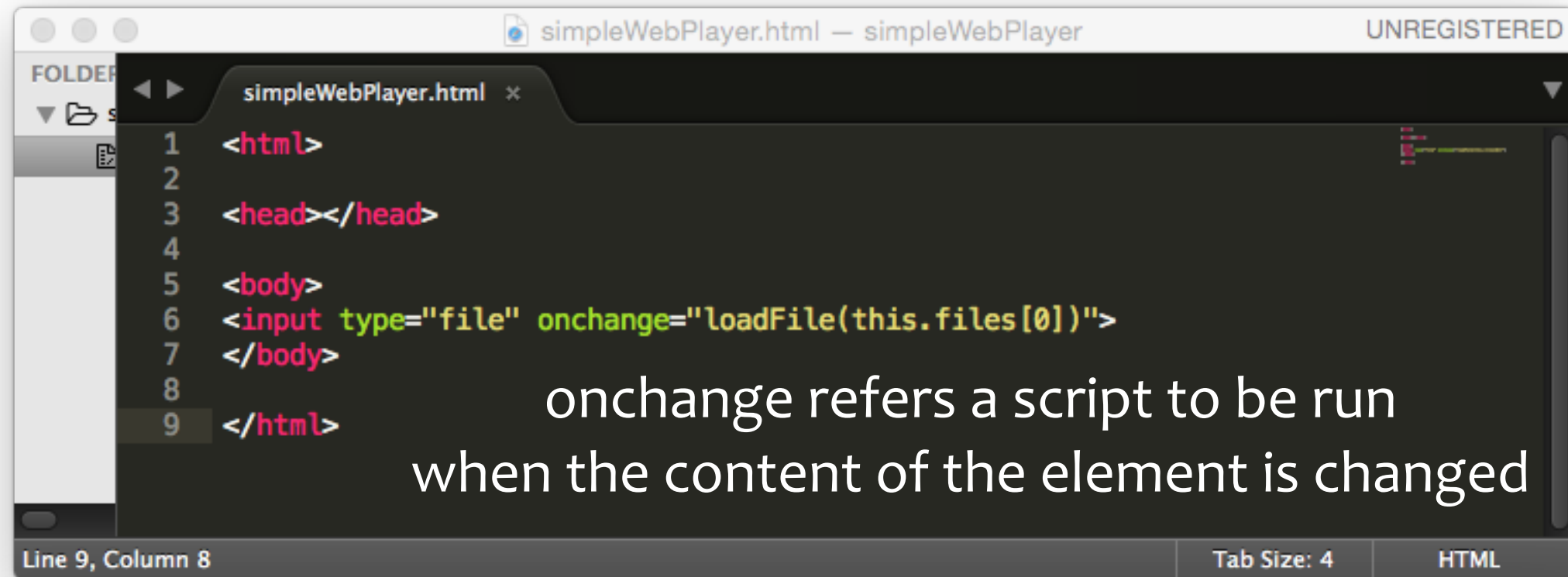
```
1 <html>
2
3 <head>
4 </head>
5 <body>
6 Hello, world!
7 </body>
8
9 </html>
```

The status bar at the bottom indicates "Line 6, Column 14", "Tab Size: 4", and "HTML".



Save and open it in your browser

Add a file chooser



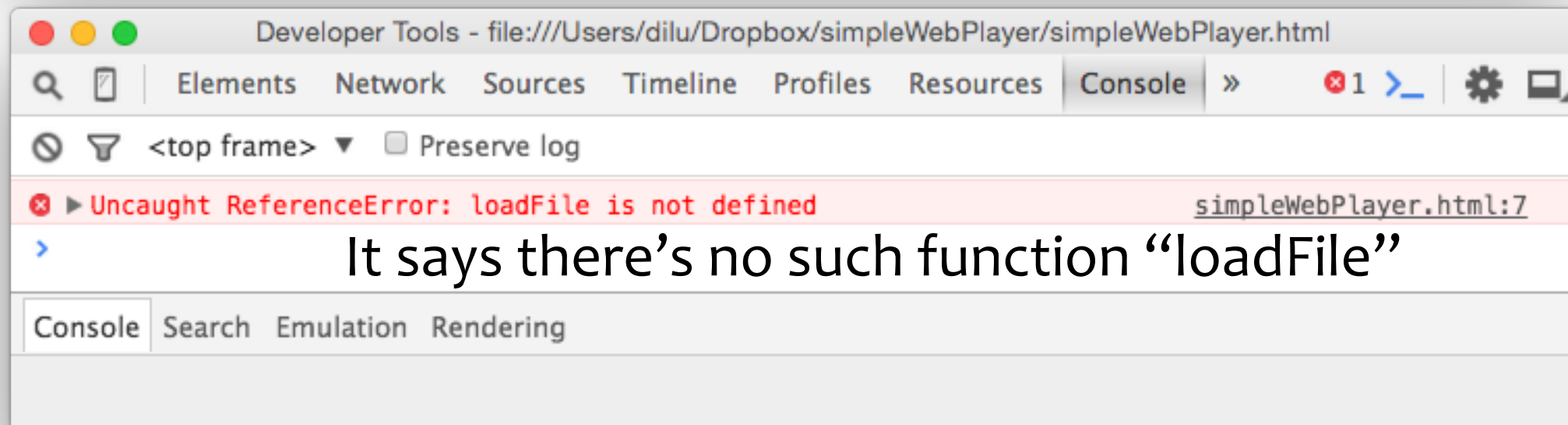
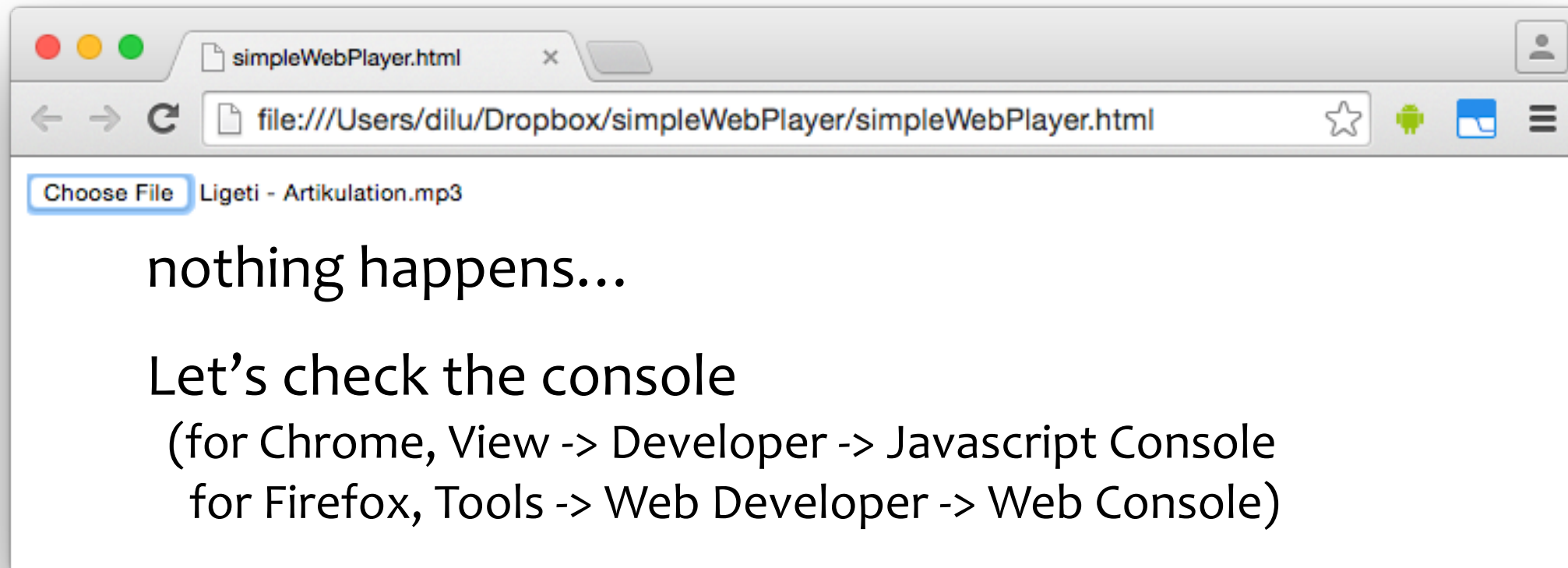
```
1 <html>
2
3 <head></head>
4
5 <body>
6 <input type="file" onchange="loadFile(this.files[0])">
7 </body>
8
9 </html>
```

onchange refers a script to be run
when the content of the element is changed

Line 9, Column 8 Tab Size: 4 HTML



Try choosing a file



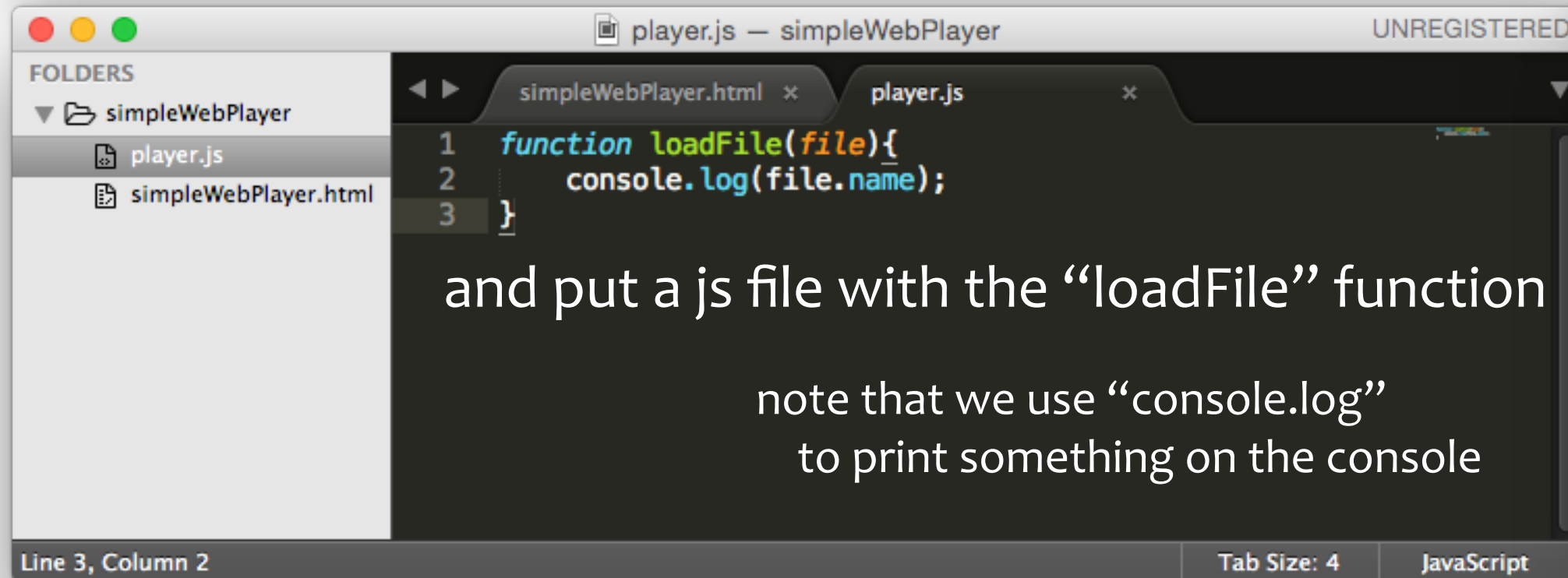
Add a Javascript file



```
1 <html>
2
3 <head>
4 <script src="player.js"></script>
5 </head>
6
7 <body>
8 <input type="file" onchange="loadFile">
9 </body>
10
11 </html>
```

add a “script” tag to import a js file

Line 5, Column 8

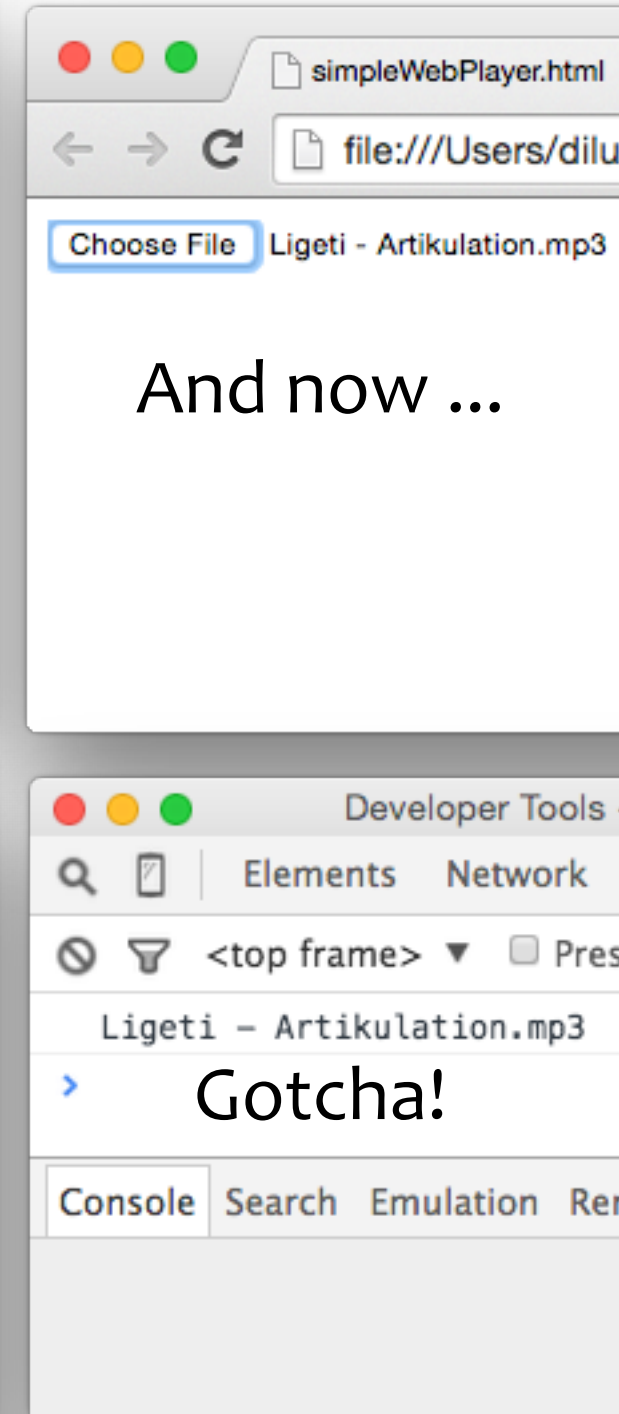


```
1 function loadFile(file){
2   console.log(file.name);
3 }
```

and put a js file with the “loadFile” function

note that we use “console.log”
to print something on the console

Line 3, Column 2



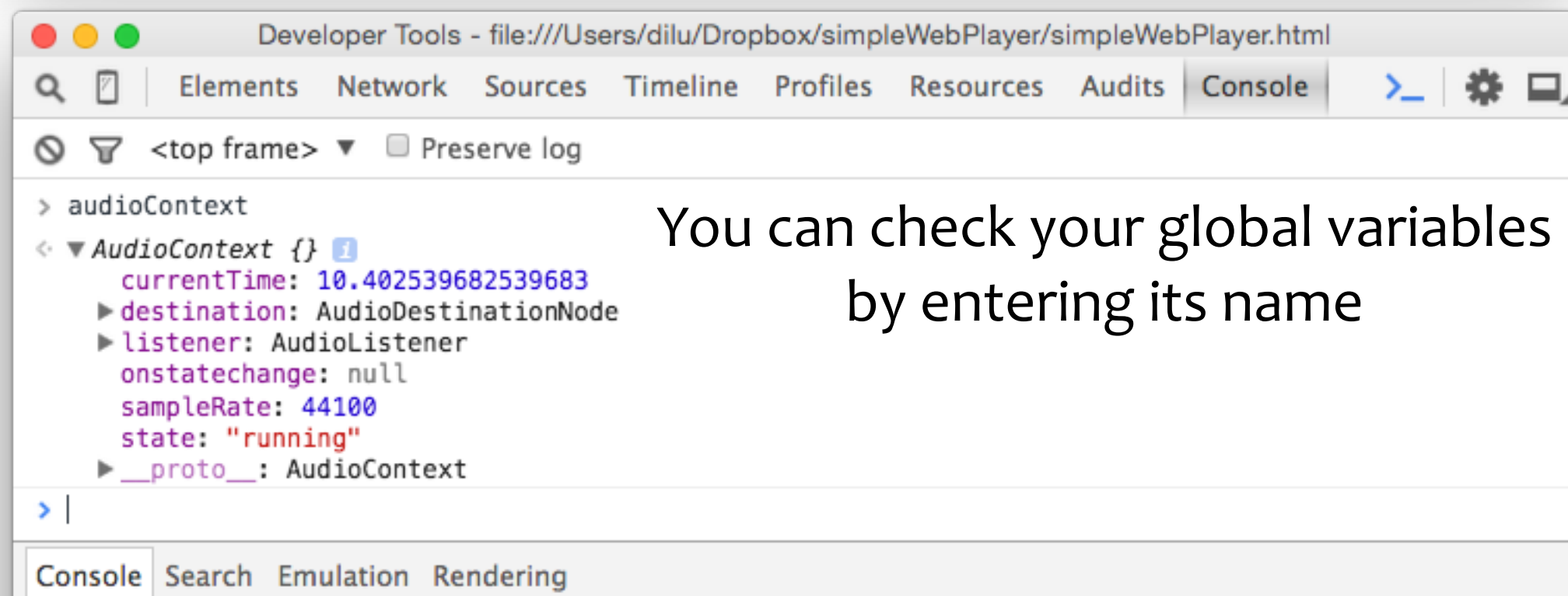
Let's create an (instance of) AudioContext



The screenshot shows a code editor window titled "player.js — simpleWebPlayer" with a status bar indicating "UNREGISTERED". The left sidebar shows a file tree with "simpleWebPlayer" containing "player.js" and "simpleWebPlayer.html". The main editor area shows the following JavaScript code:

```
1 var audioContext;
2
3 window.onload=function(){
4     audioContext = new AudioContext();
5 }
6
7 function loadFile(file){
8     console.log(file.name);
9 }
```

Annotations on the right side of the code editor state: "Declare a global variable" next to line 1, and "And after the page is loaded, create it" next to lines 3-5. The status bar at the bottom shows "Line 9, Column 2", "Tab Size: 4", and "JavaScript".

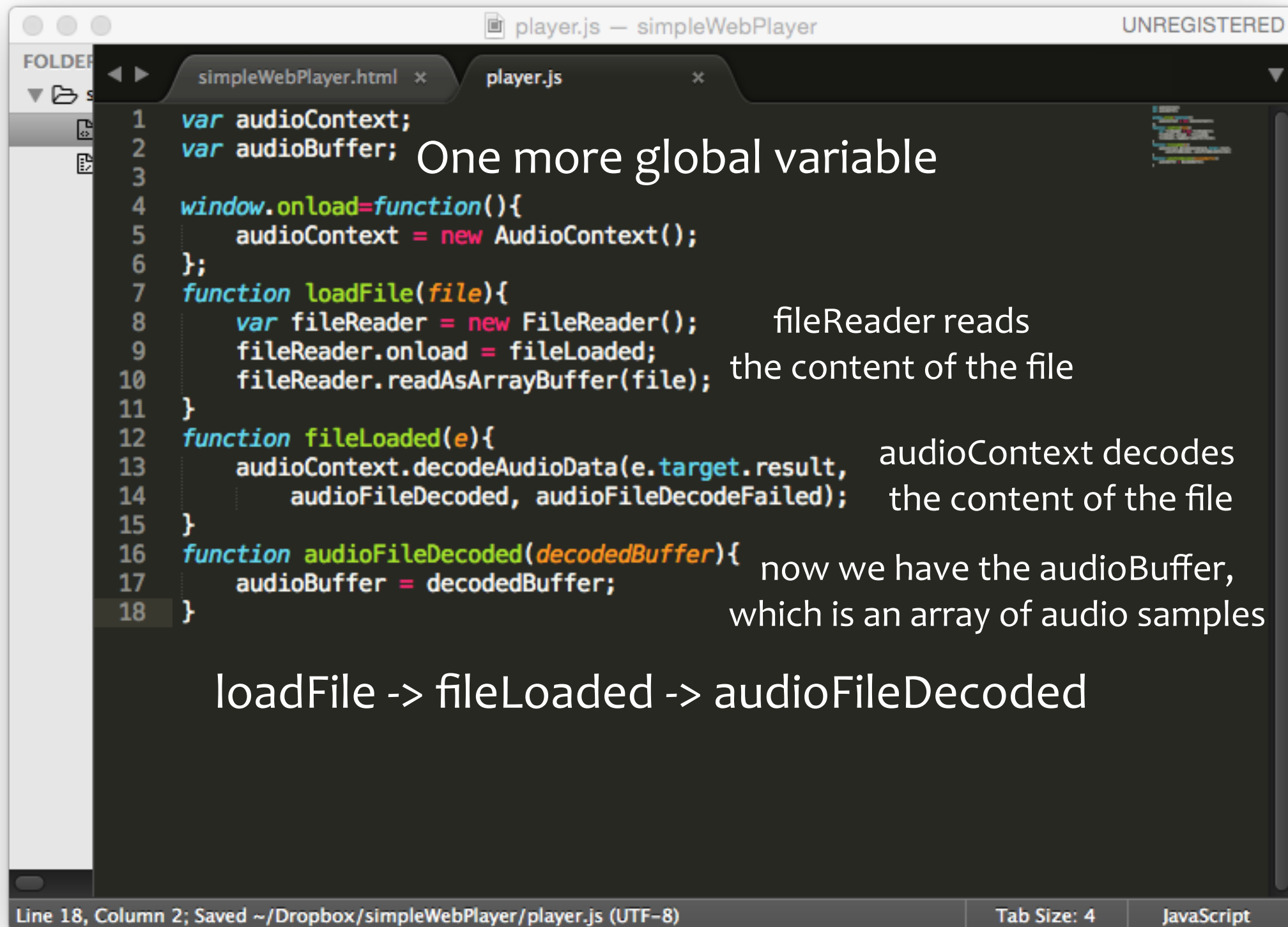


The screenshot shows the Chrome Developer Tools console for the file "file:///Users/dilu/Dropbox/simpleWebPlayer/simpleWebPlayer.html". The "Console" tab is active, showing a log entry for the "audioContext" variable. The log entry is expanded, showing the following properties:

- currentTime: 10.402539682539683
- destination: AudioDestinationNode
- listener: AudioListener
- onstatechange: null
- sampleRate: 44100
- state: "running"
- __proto__: AudioContext

Annotations on the right side of the console state: "You can check your global variables by entering its name". The bottom of the console shows tabs for "Console", "Search", "Emulation", and "Rendering".

Load the file and decode



The screenshot shows a code editor window titled "player.js — simpleWebPlayer" with a status bar indicating "UNREGISTERED". The editor has two tabs: "simpleWebPlayer.html" and "player.js". The "player.js" tab is active, displaying the following JavaScript code:

```
1 var audioContext;  
2 var audioBuffer;  
3  
4 window.onload=function(){  
5     audioContext = new AudioContext();  
6 };  
7 function loadFile(file){  
8     var fileReader = new FileReader();  
9     fileReader.onload = fileLoaded;  
10    fileReader.readAsArrayBuffer(file);  
11 }  
12 function fileLoaded(e){  
13     audioContext.decodeAudioData(e.target.result,  
14         audioFileDecoded, audioFileDecodeFailed);  
15 }  
16 function audioFileDecoded(decodedBuffer){  
17     audioBuffer = decodedBuffer;  
18 }
```

Annotations on the right side of the code explain the steps:

- Line 2: `var audioBuffer;` One more global variable
- Line 8: `var fileReader = new FileReader();` fileReader reads
- Line 10: `fileReader.readAsArrayBuffer(file);` the content of the file
- Line 13: `audioContext.decodeAudioData(e.target.result,` audioContext decodes
- Line 14: `audioFileDecoded, audioFileDecodeFailed);` the content of the file
- Line 16: `function audioFileDecoded(decodedBuffer){` now we have the audioBuffer,
- Line 17: `audioBuffer = decodedBuffer;` which is an array of audio samples

loadFile -> fileLoaded -> audioFileDecoded

Line 18, Column 2; Saved ~/Dropbox/simpleWebPlayer/player.js (UTF-8) Tab Size: 4 JavaScript

Write play()

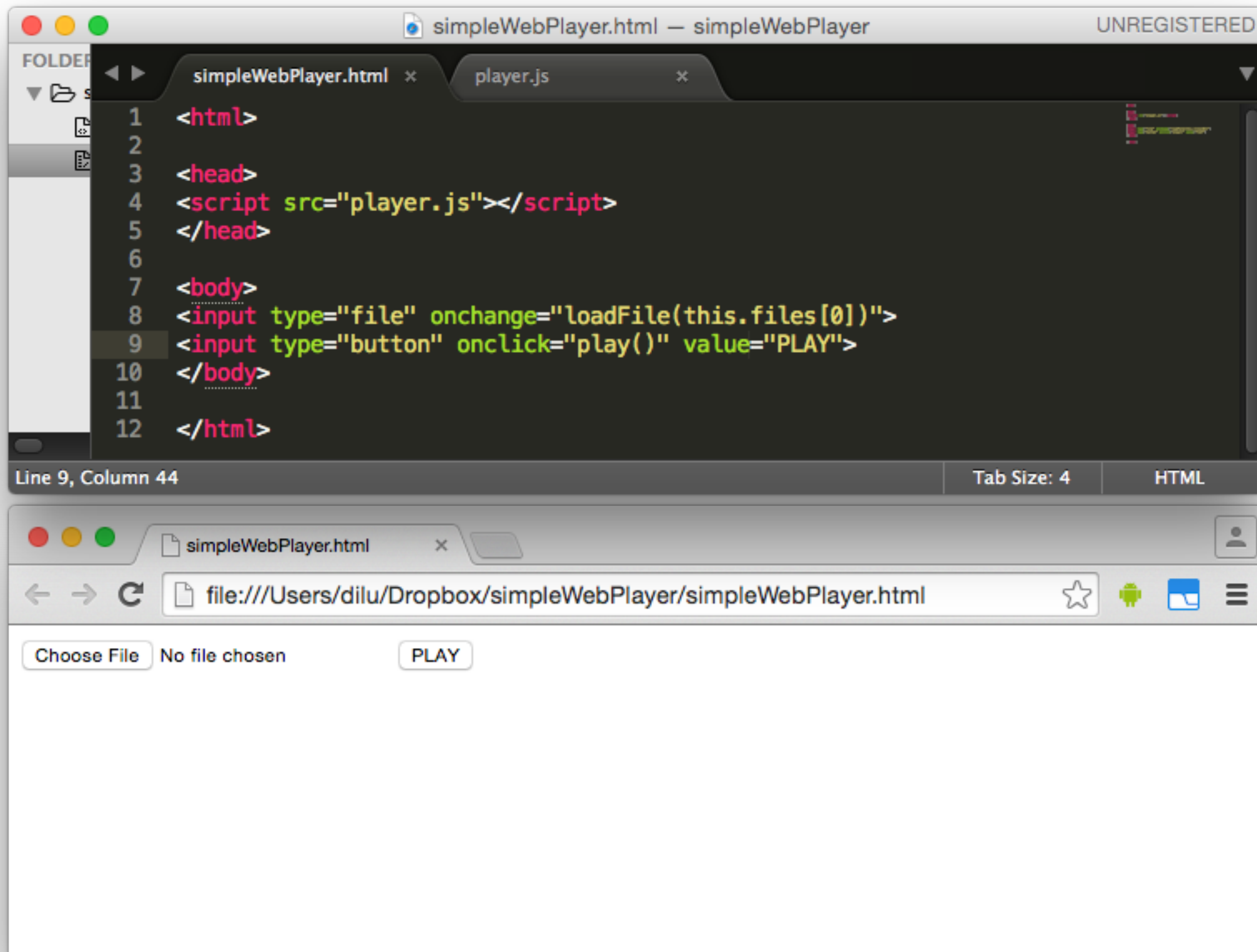
```
1  var audioContext;  
2  var audioBuffer;  
3  var bufferSourceNode; Yet another global variable  
4  
5  window.onload=function(){  
6      audioContext = new AudioContext();  
7  };  
8  function loadFile(file){  
12 }  
13 function fileLoaded(e){  
16 }  
17 function audioFileDecoded(decodedBuffer){  
18     audioBuffer = decodedBuffer;  
19 }  
20 function audioFileDecodeFailed(e){ An error handling function  
21     alert("The audio file cannot be decoded");  
22 }  
23 function play(){  
24     if(audioBuffer){ Stop the bufferSourceNode if it is playing  
25         if(bufferSourceNode) bufferSourceNode.stop();  
26         bufferSourceNode = audioContext.createBufferSource();  
27         bufferSourceNode.buffer = audioBuffer;  
28         bufferSourceNode.connect(audioContext.destination);  
29         bufferSourceNode.start(); Create a new node, set the buffer,  
30     } connect it to the destination note, and start the source  
31 }
```

Line 19, Column 2

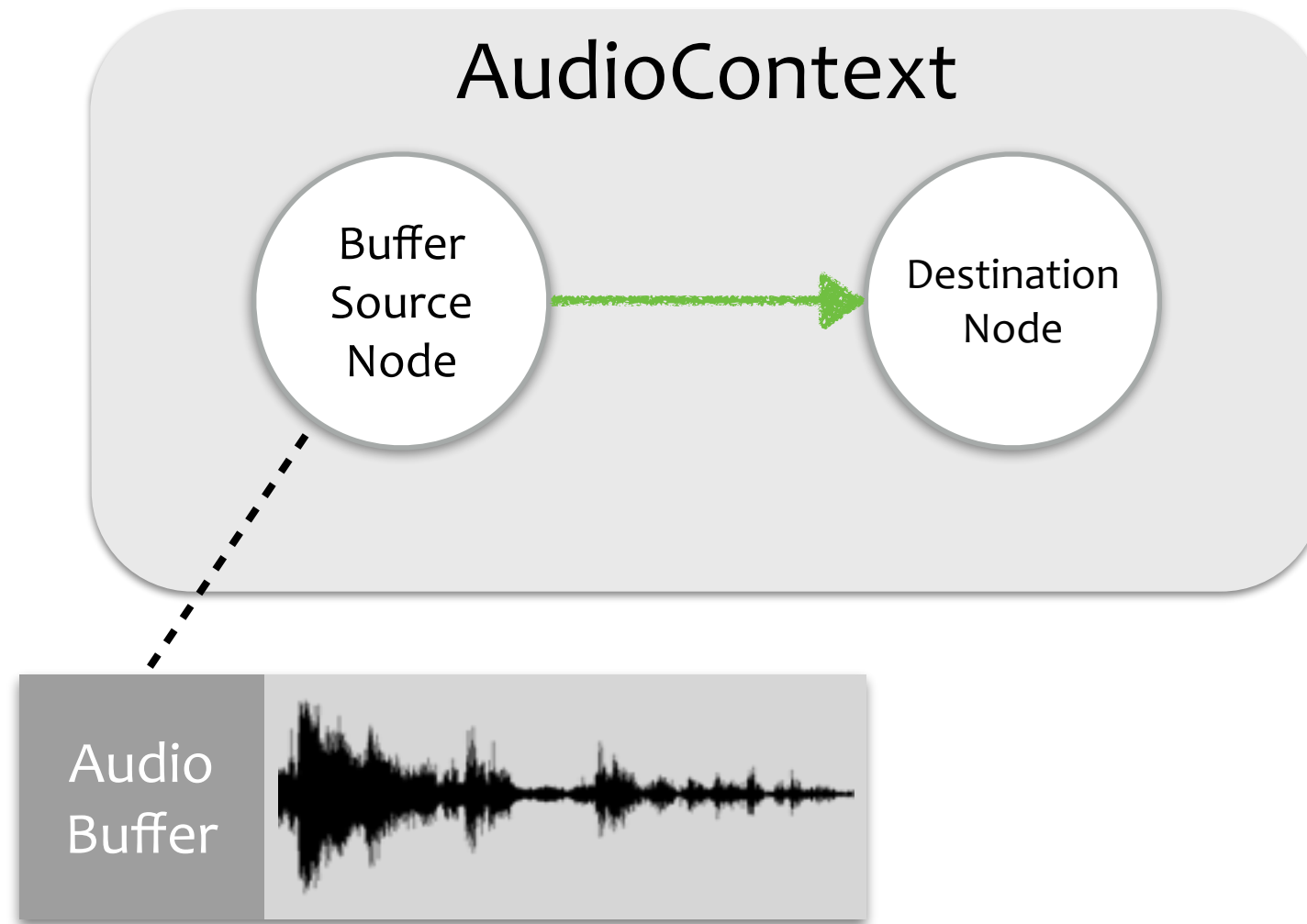
Tab Size: 4

JavaScript

Add a button for play



System Overview





“아저씨는 미리 준비해 왔어요.”

Recommended Links

- Web Audio API book (O'Reilly):
<http://chimera.labs.oreilly.com/books/1234000001552/>
- W3C's Web Audio API Reference:
<http://webaudio.github.io/web-audio-api/>
- Mozilla's Web Audio API page:
https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

Q&A