

# JavaScriptで音声生成

いで @ide\_an

<http://www.usamimi.info/~ide/>

# なんの話？

- JavaScriptで音を鳴らしたい
- それも既存のmp3ファイルとかを鳴らすんじゃなくて、その場で波形を生成して鳴らしたい

→Audio Data APIやWeb Audio APIでできるようになったよ！！！！

って話

# 昔ばなし

- HTML5登場以前
  - 音声生成はかなり無理があった
  - 既存の音声ファイルを再生する手段もアレだった
    - bgsoundやembed要素を使えばできたけどブラウザ互換性とかタイムラグの酷さとかが...
- HTML5登場以後
  - 音声ファイルの再生はaudio要素に統一(予定)
    - コーデックでモメてるけどね
  - 音声生成についてはあまり変わらなかった

# これまでであった音声生成の手段

- Flashと連携して波形データを再生させる
  - Flash10から動的に生成した波形の再生がサポートされたのでこれを使う
    - 使用例:JSNES
- audio要素とdata URIを組み合わせる
  - dataURIはバイナリファイルを文字列で表現する方法
  - 動的に生成したwaveファイルをdata URIで表現してaudio要素に飲ませて再生する
    - 使用例:jsfx,mck for javascript

# これま JavaScriptで再生と言いつつ Flashに投げてるあたりが外道 成の手段

JavaScriptで再生と言いつつ  
Flashに投げてるあたりが外道

- Flashと連携して波形データを再生させる
  - Flash10から動的に生成されたのでこれを再生できないあたりが外道
  - 使用例: JSNES
- audio要素とdata URIを組み合わせる
  - dataURIはバイナリファイルを文字列で表現する方法
  - 動的に生成したwaveファイルをdata URIで表現してaudio要素に飲ませて再生する
  - 使用例: jsfx, mck for javascript

# 新たな音声生成手段の登場

- Audio Data API
  - Firefox4で導入されたAPI
- Web Audio API
  - W3Cが策定を進めているAPI
  - Google ChromeやSafariで使える

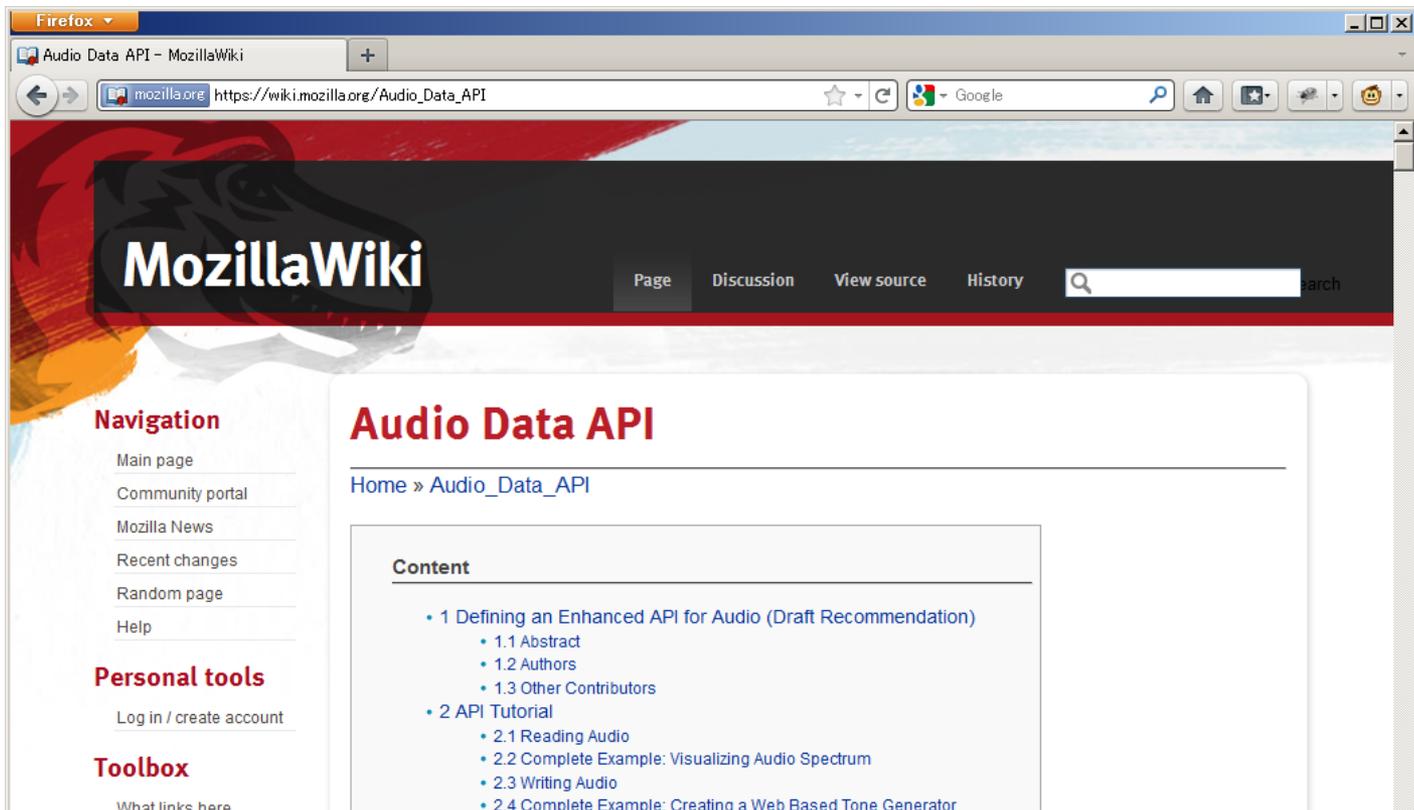




こいつらのことは忘れよう

# Audio Data API

- Firefox4で導入されたAPI
- 波形データの読み書きができる



# Audio Data APIで音声生成

## 手順

### 1. 初期化

```
var audio_output = new Audio();  
audio_output.mozSetup( ch, sample_rate );
```

### 2. 定期的に波形を書き込む

```
setInterval(function(){  
    var sample = new Float32Array( sample_length );  
    sample[0] = 0.5;  
    //・・・といった感じで波形データを作る  
    var written = audio_output.mozWriteAudio( sample );  
}, interval );
```

# Audio Data APIの長所・短所

## いい点

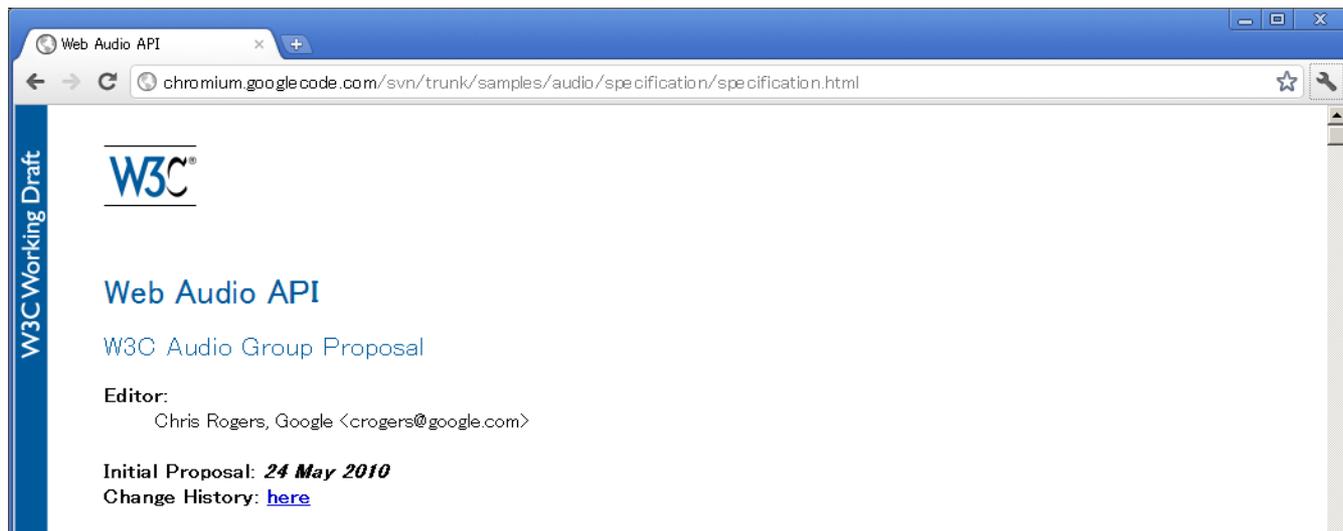
- シンプルで分かりやすい

## 悪い点

- mozWriteAudioの呼び出し間隔をプログラマが調整しないといけないのがめんどい
- mozWriteAudioに波形データがすべて書き込まれるとは限らない
  - 書き込めなかった分をあとで書き直してやらないといけない
  - 研究報告書に書いてあるサンプルを参照

# Web Audio API

- W3Cが去年の5月から策定を進めているAPI
- 対応ブラウザ
  - Google Chromeのdev版とcanary版
    - 今年の5月 (canaryは2月頃?) から対応している
  - SafariのMac版



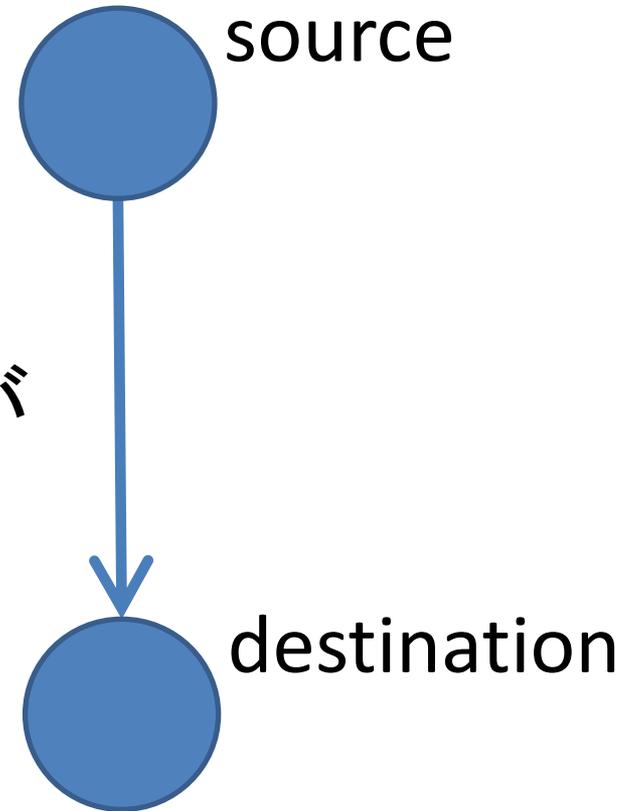
# Web Audio API

- Web Audio APIはAudio Data APIより抽象的
- 色々なエフェクタが用意されていて、それらを組み合わせて音声を加工する
- エフェクタはノードで表現され、これをつなぎ合わせることで音声加工の手順が決まる
- 他に特別なノードがある
  - sourceノード: 入力なしで波形を出力する
  - destinationノード: 入力された波形をスピーカーに出力する。常に一個だけ必要

# ノードをつなぐ(単純な例)

sourceが出力する波形を  
そのままスピーカーに出力する

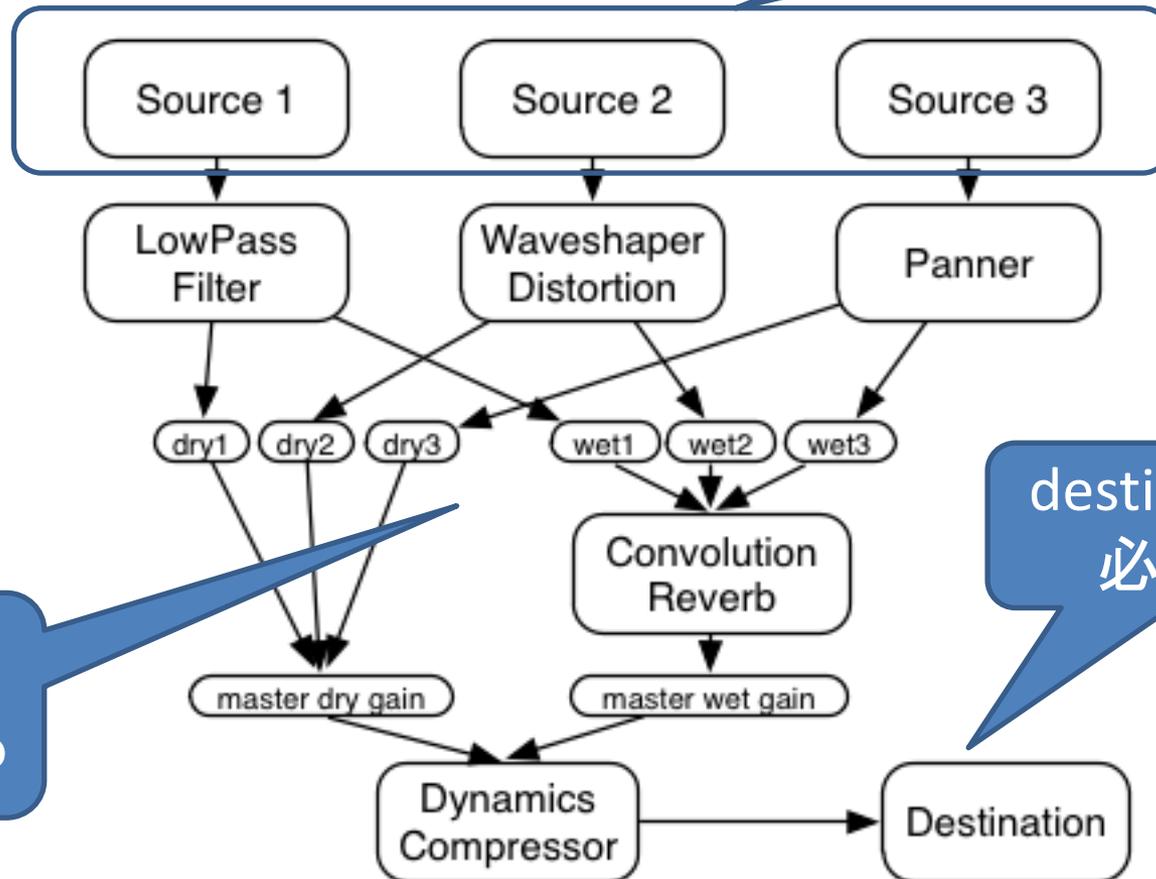
sourceノードにはいくつか種類が  
あって、JavaScriptで生成した波  
形を出力するノードもある  
(後述)



# ノードをつなぐ(複雑な例)

sourceを複数繋げる

AudioContext



エフェクタ  
色々繋げる

destinationは  
必ず1個

# Web Audio APIで音声生成

- source -> destinationのつなぎ方で実現する
  - sourceとしてJavaScriptNodeを使う

## 手順

1. 初期化
2. JavaScriptNodeに波形を生成する関数を指定
3. 各ノードをつなぐ

# Web Audio APIで音声生成

## 1. 初期化

```
var context = new webkitAudioContext();  
var js_node = context.createJavaScriptNode(  
    buffer_size , 0, 1);
```

0と1はそれぞれ入力と出力の数。

sourceとして使うので入力0,出力1

buffer\_sizeには

256,512,1024,2048,4096,8192,16384

のいずれかを指定する

# Web Audio APIで音声生成

## 2. 波形を生成するコールバック関数を設定

```
js_node.onaudioprocess = function(e){  
    var out_l = e.outputBuffer.getChannelData(0);  
    var out_r = e.outputBuffer.getChannelData(1);  
    var sample = new Float32Array( sample_length );  
    sample[0] = 0.5;  
    //・・・といった感じで波形データを作る  
    out_l.set( sample );//ステレオの左側の音  
    out_r.set( sample );// 右側の音  
};
```

波形データはbuffer\_sizeで指定した長さ分だけ書きこむ

# Web Audio APIで音声生成

## 3. 各ノードをつなぐ

```
js_node.connect( context.destination );
```

# Web Audio APIの長所・短所

- いい点
  - 波形を書き込むタイミングはAPIが調整してくれる
- 悪い点
  - サンプル周波数を自分で設定できない
    - context.sampleRateの値(48000)で固定っぽい
  - 資料が少ない
    - 仕様書だけが頼り

# 音声生成の応用

- Audio Data API/Web Audio APIを使って楽器的なものを作ってみた
- Firefox4, Google Chrome (devとcanary) で動作確認済み

# まとめ

- Audio Data API/Web Audio APIでJavaScriptによる音声生成が現実的になった
- 今後JavaScriptによる音系コンテンツが増えていくかも？
- ~~仕様が立ち消えになったりして...~~