

A white cat stands on a vibrant, multi-colored nebula background. The nebula features swirling patterns of red, orange, yellow, green, blue, and purple. The cat is positioned in the upper right quadrant, facing left. It has a fluffy white coat and is standing on its hind legs.

BACON.JS & TALKIE.JS

@ahomu / Ayumu Sato

TL;DR

Keynoteに疲れたので、自分でHTMLスライド作ったら
まあまあ満足しました。ベーコンおいしいです。^q^



Talkie.js のこと

車輪の再発明をした背景

Frontendのスタメン(?)っぽいメンバーは、ビジュアルデザインにもステータスをふっている人が多く、毎回オシャレなスライドを作ってくる。2012年当時は[Reveal.js](#)で作っていたが周りとのギャップ、ビジュアル作りに限界を感じてKeynoteに移行。以後、Keynoteでオブジェクトのサイズや配置を調整することに力を傾けた[スライド作りスタイル](#)が始まる。ところが最近になって、Web開発者なのにKeynoteを調整する日々に疑問を抱くようになりHTMLスライドに戻りたい気持ちが高まる。(イマココ)

ということで作りました

洋モノの既製品は日本語スライドと相性が微妙だった

ahomu / Talkie

 Unwatch ▾ 1 Star 10 Fork

A slide presentation library. Responsive scaling & markdown ready. <http://ahomu.github.io/Talkie/> — Edit

 49 commits 2 branches 0 releases 1 contributor Code Issues Pull Requests Wiki Pulse Graphs Settings SSH clone URL`git@github.com:ahomu/Talkie.git`

You can clone with [HTTPS](#) or [Subversion](#). 

 Clone In Desktop Download ZIP branch: master + Talkie / +

add swipe control support

ahomu authored 19 hours ago


latest commit 599e177e

add swipe control support 19 hours ago

some fixes 4 days ago

initial 21 days ago

add dist files 18 days ago

split modules 21 days ago

refactor module interface 12 days ago

update miscs 17 days ago

add swipe control support 19 hours ago

bump 0.4.2 a day ago

[index.html](#)[README.md](#)[LICENSE](#)[ulpfile.js](#)[hintrc](#)[tignore](#)[editorconfig](#)[st](#)[c](#)[st](#)[omu](#)[wip](#)[e](#)

A close-up photograph of several strips of bacon sizzling in a pan. The bacon is rendered and crispy, with visible fat and a golden-brown color. The background is dark, making the bacon stand out.

Bacon.js のこと

写真は <http://gigazine.net/news/20121018-prosit/> より

Talkie.js では
Bacon.js を利用しています

前から興味あったので使ってみた系

Bacon.js はイベントや
データの流れを
いいカンジしてくれます

Observer と Promise のパワーアップ版みたいなやつ

この3つだけ覚えればOK

- Bacon.EventStream
処理の起点になる非同期データストリーム
- Bacon.Property
ストリームから生成される現在値(状態)を示すもの
- Bacon.Bus (便利な変種ストリーム)
任意の値をpushしたり、他のStreamをplugできる

- 0 +

ふつうっぽいコード

```
// @type {Number} 現在の値
var current = 0;

// click されたら
plusEl.addEventListener('click', function(e) {
    // 加算/減算して
    current++;
    // 反映する
    counterEl.innerHTML = current;
});

minusEl.addEventListener('click', function(e) {
    current--;
    counterEl.innerHTML = current;
});
```

ベーコンがおいしいコード

```
// @type {Bacon.EventStream} +1/-1のストリーム
var plus = Bacon.fromEventTarget(pEl, 'click').map(1);
var minus = Bacon.fromEventTarget(mEl, 'click').map(-1);
var src = Bacon.mergeAll(plus, minus);

// @type {Bacon.Property} 現在の値
var current = src.scan(0, function(acc, v) {
    return acc + v;
});

// @type {Bacon.Bus} やってきたvalueをDOMに反映するバス
var counterBus = new Bacon.Bus();
counterBus.onValue(function(v) {
    counterEl.innerHTML = v;
});

counterBus.plug(current); // it works!
```

```
/*
+-----+
| click.map(1) | - - >
+-----+
+-----+
| click.map(-1)| - - >
+-----+
*/
var plus  = Bacon.fromEventTarget(pEl,'click').map(1);
var minus = Bacon.fromEventTarget(mEl,'click').map(-1);
```

```
/*
+-----+
| click.map(1) |
+-----+-----+
|
+----->
|
+-----+
| click.map(-1) |
+-----+
*/
var src = Bacon.mergeAll(plus, minus);
```

```
/*
+-----+
| click.map(1) |
+-----+-----+
|                         /-----\
|                         | scan(0, add) |
|                         \-----/
+-----+-----+
| click.map(-1) |
+-----+-----+
*/
var current = src.scan(0, function(acc, v) {
    return acc + v;
});
```

```
/*
+-----+
[bus]--->|innerHTML = v|
+-----+
*/
var counterBus = new Bacon.Bus();
counterBus.onValue(function(v) {
    counterEl.innerHTML = v;
});
```

```
/*
+-----+
| click.map(1) |
+-----+-----+
|           /-----\
+----->| scan(0, add)|-[bus]->|innerHTML = v|
|           \-----/
+-----+-----+
| click.map(-1) |
+-----+-----+
*/
var release = counterBus.plug(current);

// release(); -> release plug
```

ベーコンのおいしさ

- 手続きを書く代わりに、データの流れと関係を宣言できる
- イベントも非同期処理も値も、ストリームとして均一に扱える
- 自動で変化する Property に「状態」を封じ込められる
- 関数型っぽく、処理の粒度を小さくしたスタイルで書ける
- DOMへの副作用は処理の出口 (onValue) に集約できる

Talkie.js 中の Bacon.js 使用例

サンプルコードの陳列

ポインター

```
var move = Bacon.fromEventTarget(el, 'mousemove');

var x = move.map(function(e) { return e.x; });
var y = move.map(function(e) { return e.y; });

x.onValue(function(v) { pointer.style.left = v; });
y.onValue(function(v) { pointer.style.top =v; });
```

進む・戻る

```
var paging = require('paging');

var keyup = Bacon.fromEventTarget(doc, 'keyup');
var right = keyup.filter(keyCodeIs('right'));
var left = keyup.filter(keyCodeIs('left'));

paging.nextBus.plug(right);
paging.prevBus.plug(left);

window.next = function() { paging.nextBus.push(); }
window.prev = function() { paging.prevBus.push(); }
```

スワイプ[°] (1/2) (簡略サンプル)

```
var start = Bacon.fromEventTarget(el, 'touchstart');
var move  = Bacon.fromEventTarget(el, 'touchmove');
var end   = Bacon.fromEventTarget(el, 'touchend');

var moving = start.flatMap(function(init) {
    return move.takeUntil(end)
        .scan({}, function(acc, move) {
            return {
                init : acc.init || init.changedTouches[0],
                curt : move.changedTouches[0]
            };
        }).skip(1);
});
```

スワイプ (2/2) (簡略サンプル)

```
var swipeLeft = moving.filter(function(moves) {  
    var init = moves.init;  
    var curt = moves.curt;  
    var delta = init.clientX - curt.clientX;  
  
    return delta > 50;  
});  
  
swipeLeft.onValue(function() {  
    // swipe left!  
});
```

React と組み合わせたサンプル

github.com/ahomu/demo-react-bacon

詳しくは週末の Frontend で!!

Room A 13:30~ "Reactive Programming in JavaScript"

感想

- Bacon.js + Backbone.js/Reactあたりはアリ
- オブジェクト間のメッセージングを使うのが導入に◎
- ストリームの入出力をドキュメントで明示しないと大変
- 高階関数の記号 or 命名規則を考えないと大変
- 副作用がある関数の記号 or 命名規則を考えないと大変

A close-up photograph of an orange tabby cat's face, looking slightly upwards and to the right. The background is a vibrant, multi-colored nebula or galaxy, featuring shades of blue, green, yellow, and red, with numerous small white stars scattered across it.

QUESTION?

aho.mu

@ahomu