

LLoT Night - フロントエンドだめ自慢

RIOT

河村 奨 (かわむらつとむ)

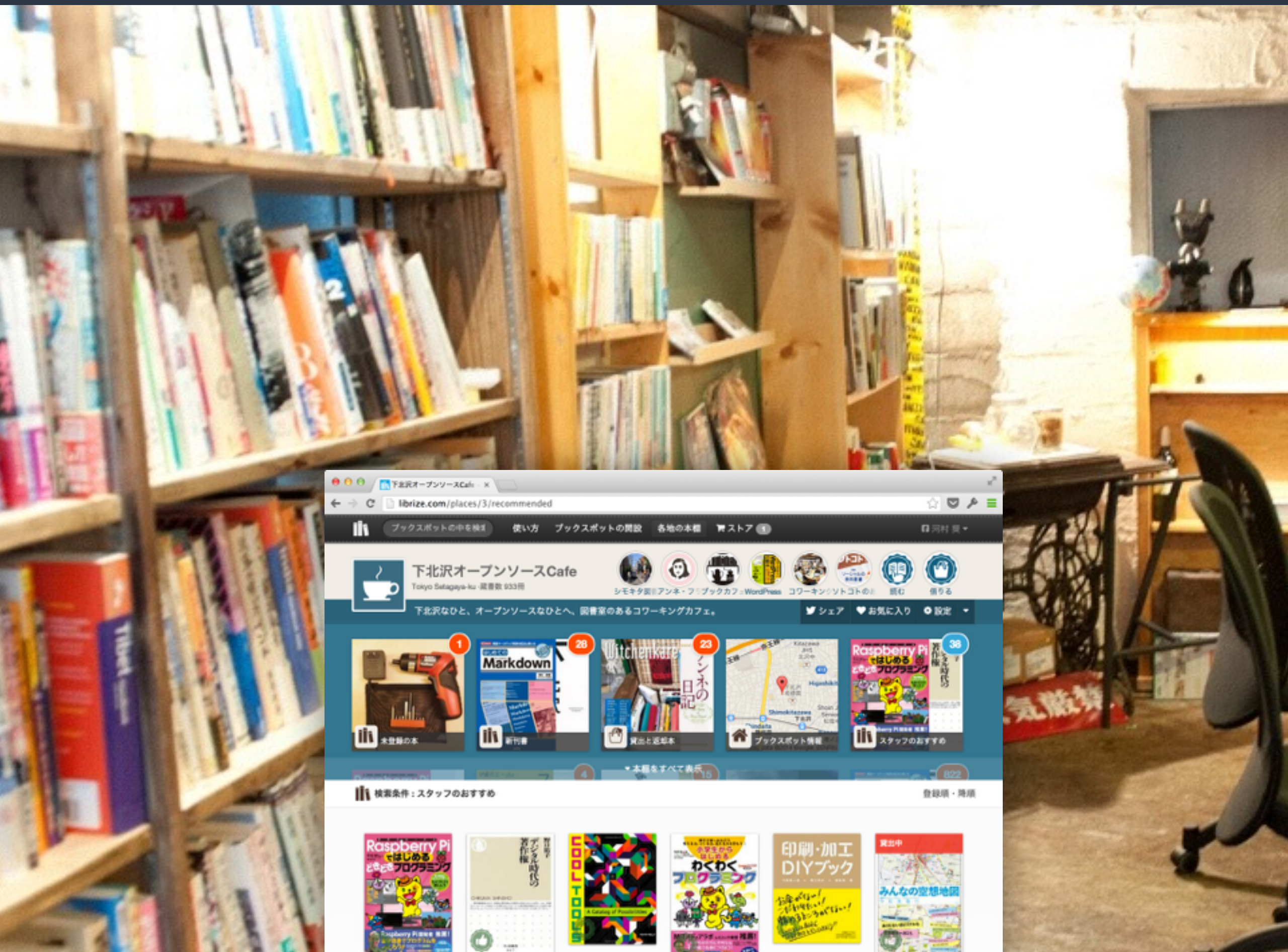
リブライズ合同会社 / 下北沢オープンソースCafe



OpenSource Cafe
• Mon-Thu 10am-8pm
• Fri Closed
• Sat-Sun 10am-8pm
Cafe & Library for Coding,
OpenSource Activities, and more!!

OpenSource Cafe

下北沢 オープンソース・カフェ by CogniTom



リブライズ

CoderDojo



cognitom

github.com/cognitom

The screenshot shows the GitHub interface for the 'Riot' repository. The browser tab is titled 'Riot' and the address bar shows 'https://github.com/riot'. The repository name 'Riot' is displayed in large text, with the description 'A React- like, user interface library' and the website 'http://riotjs.com'. The repository has 10 people and 2 teams. A green button '+ New repository' is visible. The 'People' section shows a grid of avatars, with a circular callout highlighting a subset of them.

Riot

A React- like, user interface library

<http://riotjs.com>

Repositories

People 10 Teams 2 Settings

Filters Find a repository...

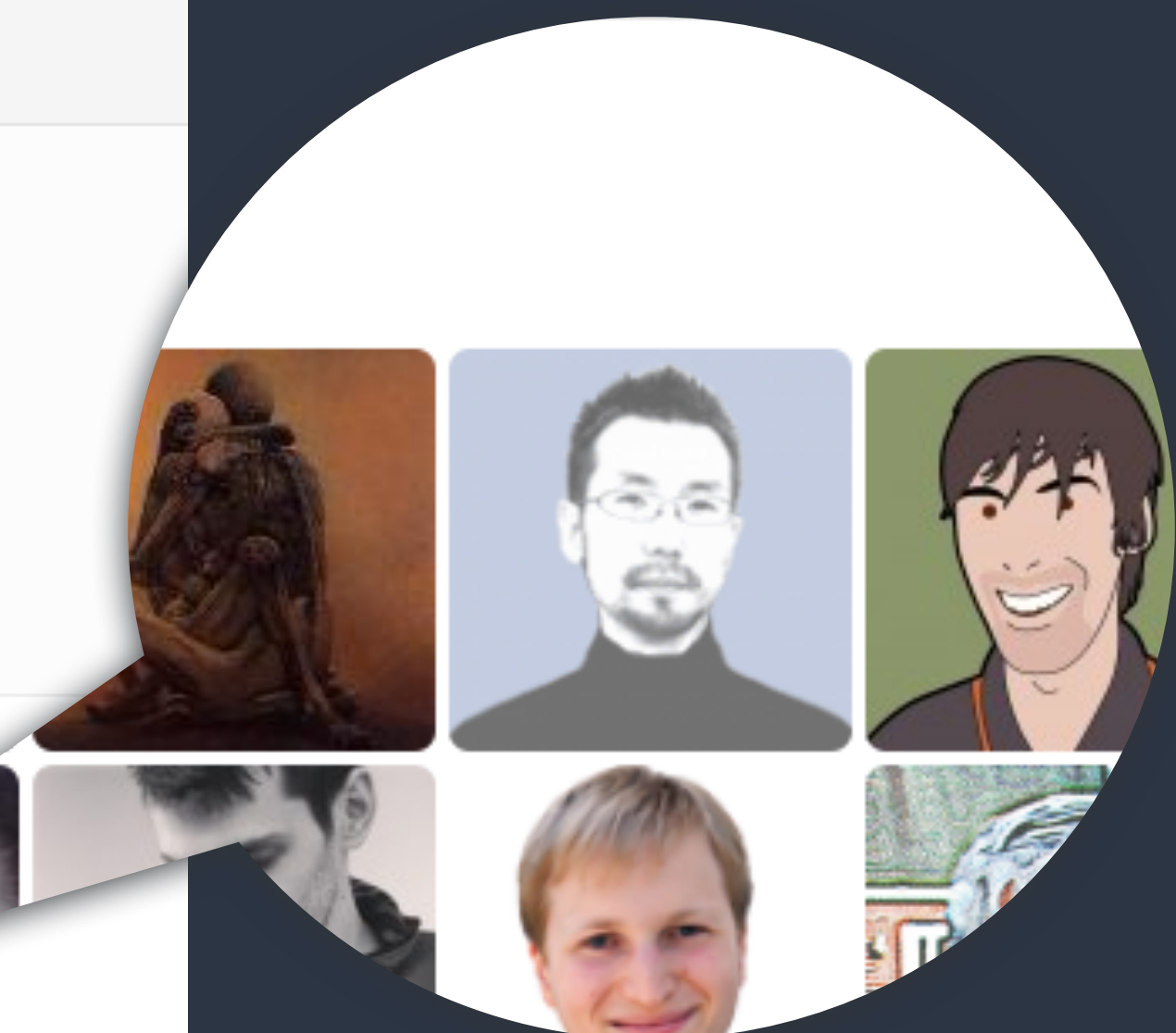
+ New repository

HTML ★ 30 3

examples

Demos and examples for Riot and submodules

Updated 2 hours ago



Riot とは

RIOT

シンプルで、とにかく軽い UIライブラリ



小さいことは、善

3. サイズが小さい

polymer.min.js – 53.0KB



react.min.js – 38.56KB



riot.min.js – 8.61KB



- 少ないバグ
- パースが早く、ダウンロードも容易
- エンベッダブル(組込可能): ライブラリはアプリケーション本体より小さくあるべき
- メンテナンスの手間が少ない: Riotのメンテナンスのために大きなチームを必要としない

	v2.0	v3.0
ファイルサイズ	2.5KB	9.3KB
公式リポジトリ	1個	13個
コアコミッター	2人	6人

若干ぽっちゃり 🤪

RIOT

GitHubのスター	★10,250
コントリビュータ	140人
プルリクエストの数	655件
Qiitaの記事	82件

Riotのひろがり

RIOT

コンポーネントを作れます

```
<weather type="clear" temperature="10" humidity="30%" />
```

```
<div>
  
  <ul>
    <li>空もよう: 晴れ</li>
    <li>気温: 10度</li>
    <li>湿度: 30%</li>
  </ul>
</div>
```



RIOT

**Riot.js はどんな人に
支持されているか？**

RIOT

HTML・JS・CSSを、普通に理解していて

「説明書読まない人」

RIOT

ダメ我慢 ①

RIOT



「コミッター、ゴビ砂漠に消える」

RIOT



tipiirai commented on Jun 10, 2015

Riot member



Awesome!

So glad to see this. The current `tmpl` is written by @**aurri** who is wondering around somewhere in Asian desert. Nice to see someone taking over the cryptic side of Riot and based on the comments it looks like you know what you are doing. And I like the idea of generating pure JS from tag definitions. Good stuff!

Can you brief us on higher lever what was the main issue on the current `tmpl` and what are the benefits of the new one?

wondering around somewhere in Asian desert

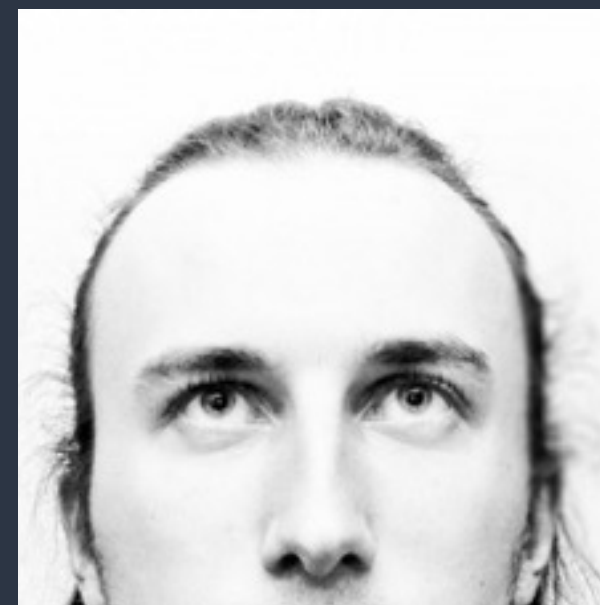
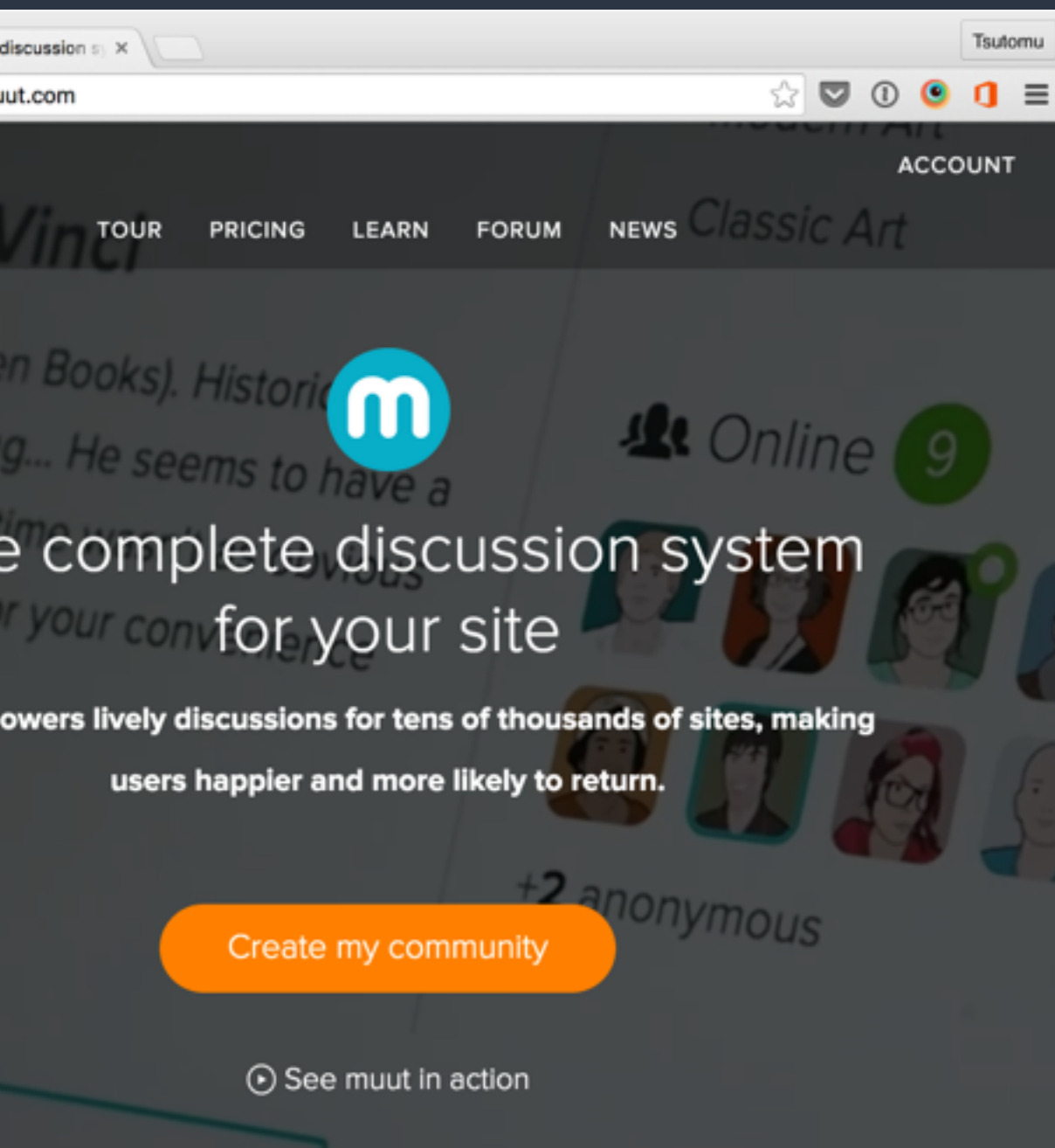
証言: 「アジアの砂漠をさまよっている」

RIOT



コアコミッターの居場所

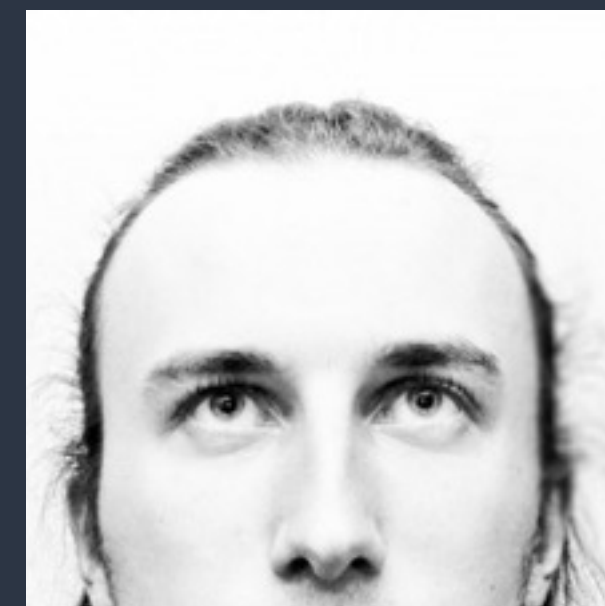
フィンランド



2015年1月 v2.0

RIOT

アジアの砂漠へ



スイスから

2015年春

RIOT



会長職的な



テキサスから



東京から



メキシコから

2015年夏

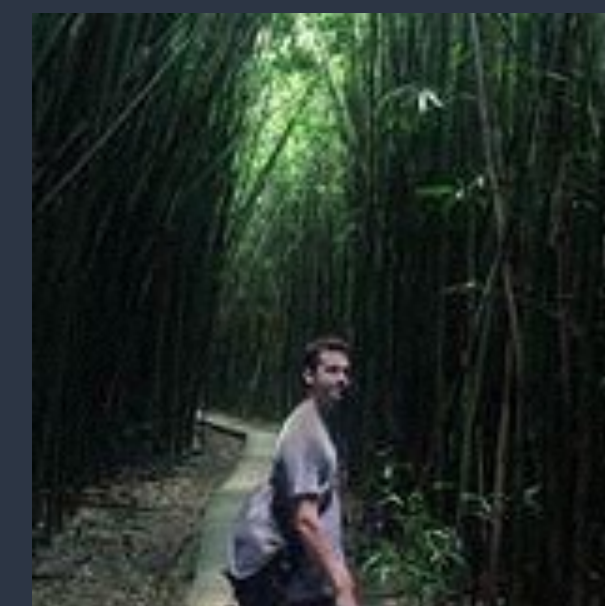




時々降臨



どこから



2016年

RIOT

ま、オープンソースですから!

RIOT

ダメ自慢 ②

RIOT

/コンパイラが正規表現/

RIOT

つまり...

RIOT

```
var HTML_TAGS =  
/<(-?[A-Za-z][-\\w\\xA0-  
\\xFF]*)?(?:\\s+([^\"]\\/>]*(?:  
(?:\"[^\"]*\"|'[^']*'|\\/[^\>]*))  
>]*)*)|\\s*)(\\/?)>/g
```

動くんだからいいじゃないか!

RIOT

という開き直り 50%

+

ASTを抽出する必要がない、という事情

```
<my-tag>  
  <p>{ message }</p>  
  <script>  
    this.message = "Hello!"  
  </script>  
</my-tag>
```



```
riot.tag2('my-tag',  
  '<p>{ message }</p>'  
  , '' , '' ,  
  function(opts) { this.message = "Hello!" }  
)
```

RIOT

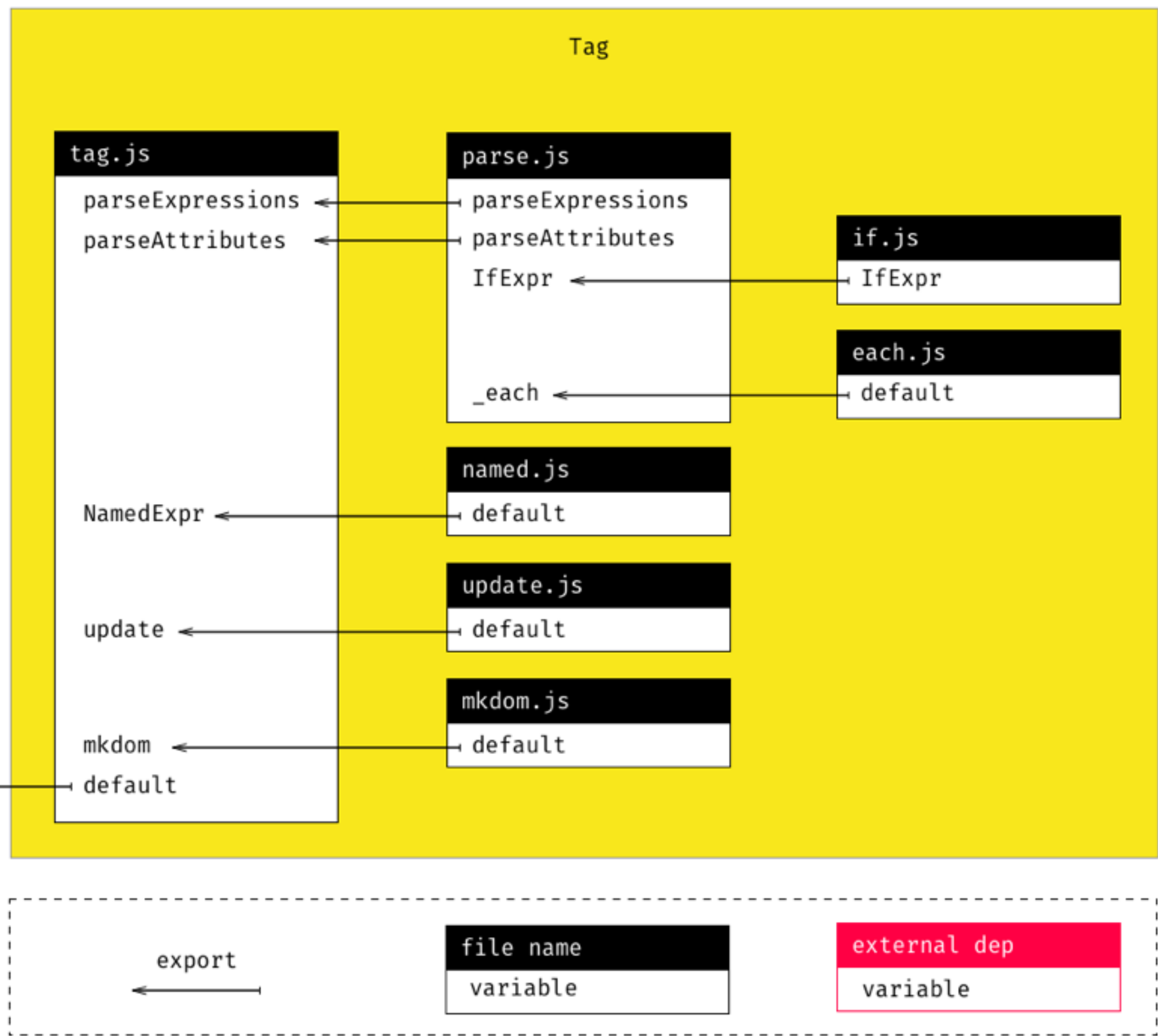
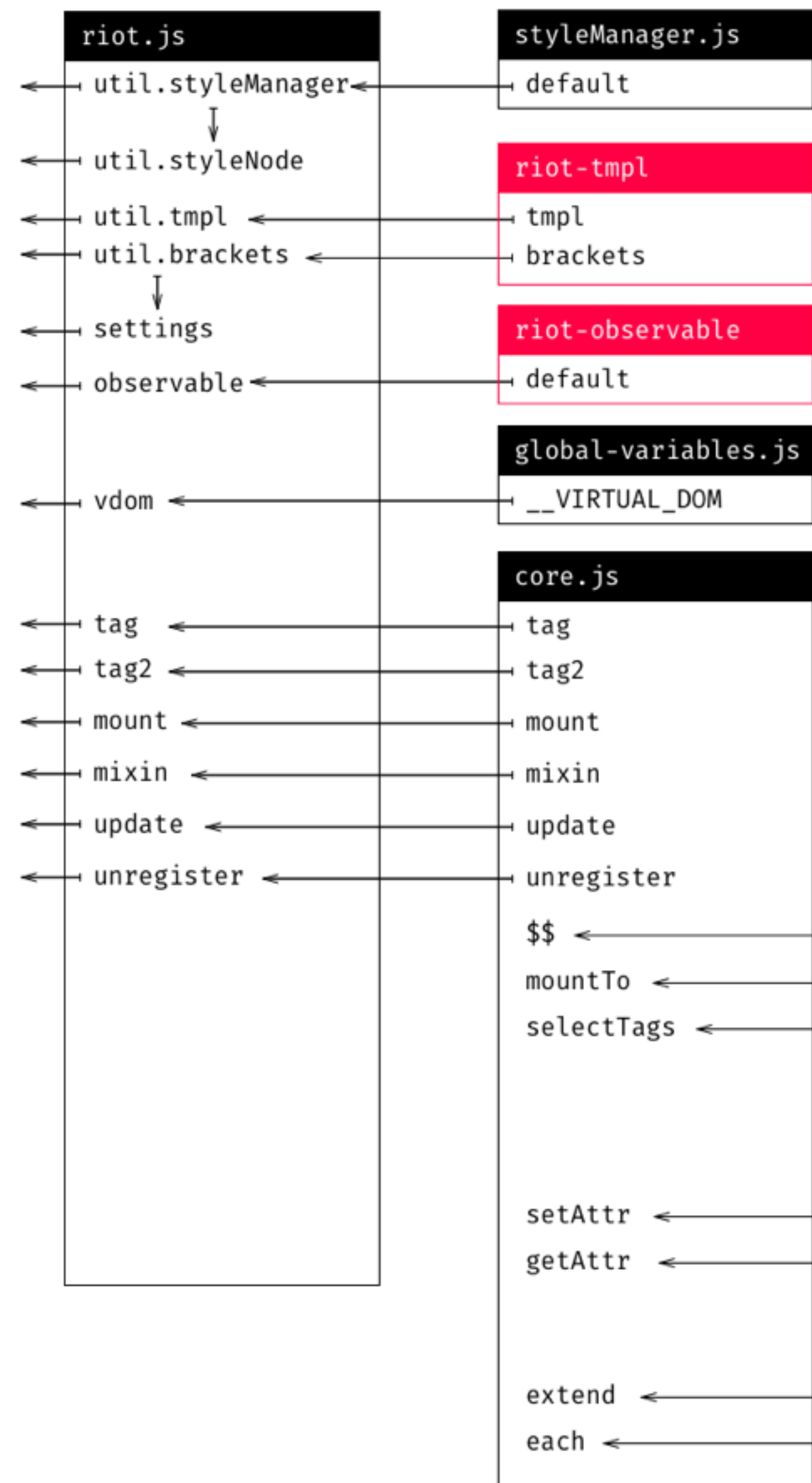
ダメ自慢 ③

RIOT

「パーサがないっ!」

あるいは、innerHTML最強説…

RIOT



Basic parts of **Riot.js 3.x**

```
core.js
function mount(selector, tagName, opts) {
  var tag = mountTo(root, riotTag
}
```

```
tag-helpers.js
function mountTo(root, tagName, opts) {
  var tag = new Tag(tag, conf, innerHTML)
  tag.mount(true)
}
```

```
tag.js
var expressions = [],
    dom = mkdom(impl.tmpl, innerHTML)

function tagMount(forceUpdate) {
  parseAttributes(root, root.attributes,
    parent, (attr, expr) => { ... })
  parseExpressions(dom, self, expressions, false)
  self.update(item)
  root.appendChild(frag)
}

function tagUpdate(data) {
  update(expressions, self)
}
```

```
update.js
function update(expressions, tag) {
  each(expressions, (expr, i) => {
    var value = tmpl(expr.expr, tag)
    if (isFunction(value))
      setEventHandler( ... )
    setAttr(dom, attrName, value)
  })
}
```

```
mkdom.js
function mkdom(templ, html) {
  var el = mkEl(GENERIC, isSVGTag(tagName))
  setInnerHTML(el, templ)
  return el
}
```

```
dom-helpers.js
function setInnerHTML(container, html) {
  container.innerHTML = html
}
```

```
parse.js
function parseAttributes(dom, attrs, tag, fn) {
  each(attrs, attr => {
    if (tmpl.hasExpr(attr.value)) {
      expr = { dom, expr: attr.value, attr: attr.name, bool }
    }
    fn(attr, expr)
  })
}

function parseExpressions(root, tag, expressions, includeRoot) {
  walkNodes(root, (dom, ctx) => {
    if (dom.nodeType === 3 && tmpl.hasExpr(dom.nodeValue)) {
      parent.children.push({dom: dom, expr: dom.nodeValue})
    }
    if (attr = getAttr(dom, 'each')) {
      parent.children.push(_each(dom, tag, attr)); return false
    }
    if (attr = getAttr(dom, 'if')) {
      parent.children.push(new IfExpr(dom, tag, attr)); return false
    }
    var tagImpl = getTag(dom)
    if (tagImpl) {
      parent.children.push(
        initChildTag(tagImpl, conf, dom.innerHTML, tag)
      ); return false
    }
    parseAttributes( ... )
    return {parent: parent}
  }, base)
}
```

```
each.js
function _each(dom, parent, expr) {
  /*
  A
  little
  bit
  complicated
  logics
  to
  make
  array
  of
  tags
  */
}
```

```
if.js
function IfExpr(dom, parentTag, expr) {
  if (newValue && !this.current) {
    this.stub.parentNode.insertBefore( ... )
  }
  else if (!newValue && this.current) {
    this.current.parentNode.removeChild( ... )
  }
}
```

The codes above shows only the most essential lines of them.
This explains the basic flow while mounting the tags in Riot.

Mounting tags in **Riot.js 3.x**

core.js

```
function mount(selector, tagName, opts) {  
  var tag = mountTo(root, riotTag  
}
```

tag-helpers.js

```
function mountTo(root, tagName, opts) {  
  var tag = new Tag(tag, conf, innerHTML)  
  tag.mount(true)  
}
```

tag.js

```
var expressions = [],  
    dom = mkdom(impl.tmpl, innerHTML)  
  
function tagMount(forceUpdate) {  
  parseAttributes(root, root.attributes,  
    parent, (attr, expr) => { ... })  
  parseExpressions(dom, self, expressions, false)  
  self.update(item)  
  root.appendChild(frag)  
}
```

mkdom.js

```
function mkdom(templ, html) {  
  var el = mkEl(GENERIC, isSVGTag(tag  
  setInnerHTML(el, templ)  
  return el  
}
```

parse.js

```
function parseAttributes(d  
  each(attrs, attr => {  
    if (templ.hasExpr(attr.  
      expr = { dom, expr:  
    }  
    fn(attr, expr)  
  })  
}  
  
function parseExpressions(  
  walkNodes(root, (dom, ct  
    if (dom.nodeType = 3  
      parent.children.push  
    }  
    if (attr = getAttr(dom  
      parent.children.push  
    }  
    if (attr = getAttr(dom  
      parent.children.push  
    }  
  }
```

dom.js

```
function mkdom(templ, html) {  
  var el = mkEl(GENERIC, isSVGTag(tagName))  
  setInnerHTML(el, templ)  
  return el  
}
```

dom-helpers.js

```
function setInnerHTML(container, html) {  
  container.innerHTML = html  
}
```

parse.js

```
function parseAttributes(dom, attrs, tag, fn) {  
  each(attrs, attr => {  
    if (templ.hasExpr(attr.value)) {  
      expr = { dom, expr: attr.value, attr: attr }  
      fn(attr, expr)  
    }  
  })  
}  
  
function parseExpressions(root, tag, expressions, includeRoot) {  
  walkNodes(root, (dom, ctx) => {  
    if (dom.nodeType === 3 && templ.hasExpr(dom.nodeValue)) {  
      parent.children.push({dom: dom, expr: dom.nodeValue})  
    }  
  })  
}
```

each.js

```
function _each(dom, fn) {  
  /*  
   * A  
   * little  
   * bit  
   * complicated  
   * logics  
   * to  
   * make  
   */  
}
```

あ！

マウントするまで

- ・ `riot.tag2()`関数で登録済みの、タグの実装を取得
- ・ タグをインスタンス化
- ・ タグの実装(HTMLテンプレート)をinnerHTMLに突っ込む
- ・ **パースする、というよりDOMをトラバースして、解析**
- ・ 与えられたデータで、タグの内容を更新
- ・ テンプレート変数(expression)を更新
- ・ マウント完了イベント

RIOT

```
<my-tag>
  <p>{ message }</p>
  <script>
    this.message = "Hello!"
  </script>
</my-tag>
```



```
riot.tag2('my-tag',
  '<p>{ message }</p>',
  function(opts) { this.message = "Hello!" }
)
```

```
riot.mount('#main', 'my-tag')
```

<p>
{ message }
</p>

<p>
Hello!
</p>

#main
<p>
Hello!
</p>

1

2

3

RIOT

innnerHTMLサイキョウデスヨネ (棒)

RIOT

まとめ?

RIOT



ブラウザゆとり世代

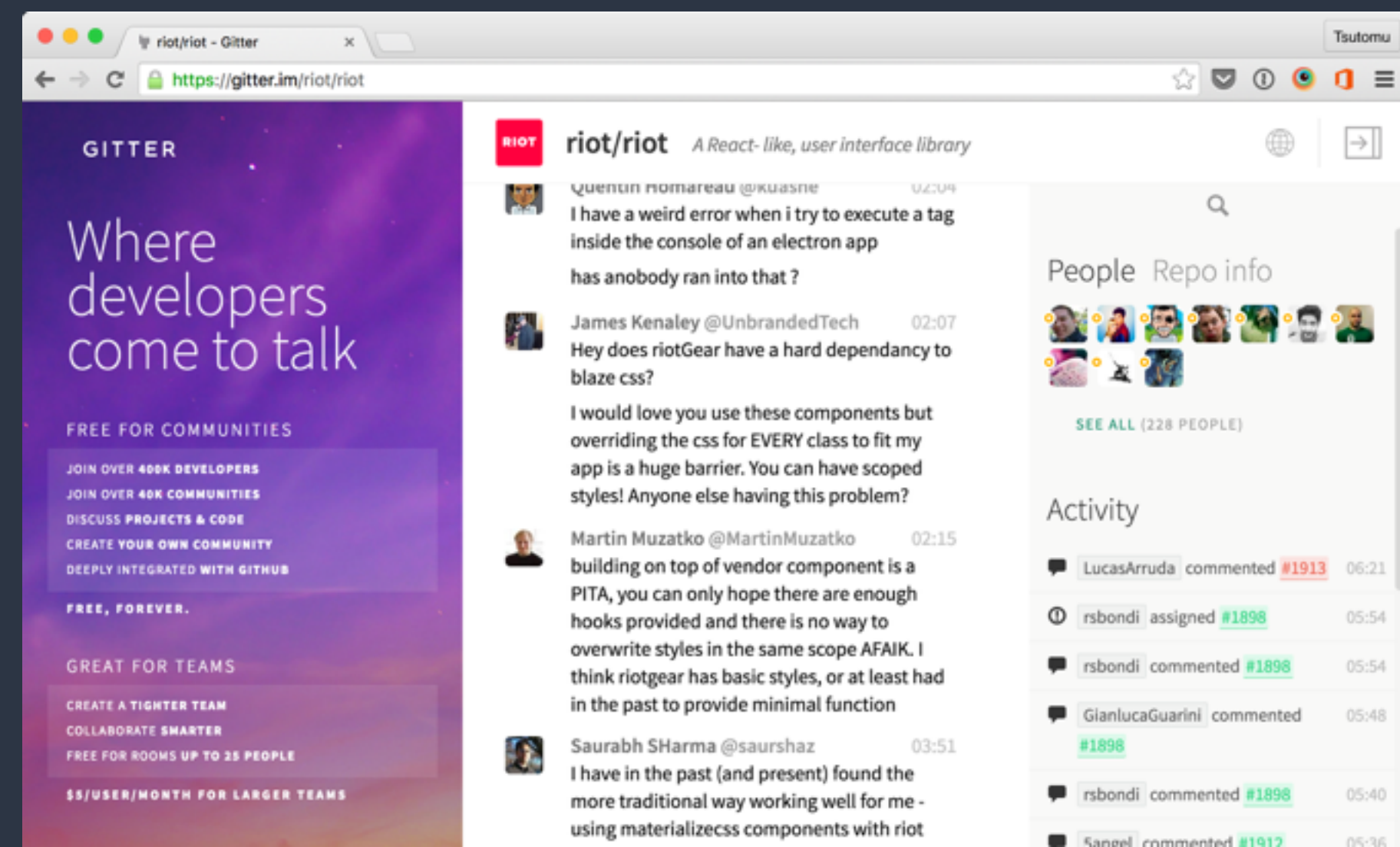
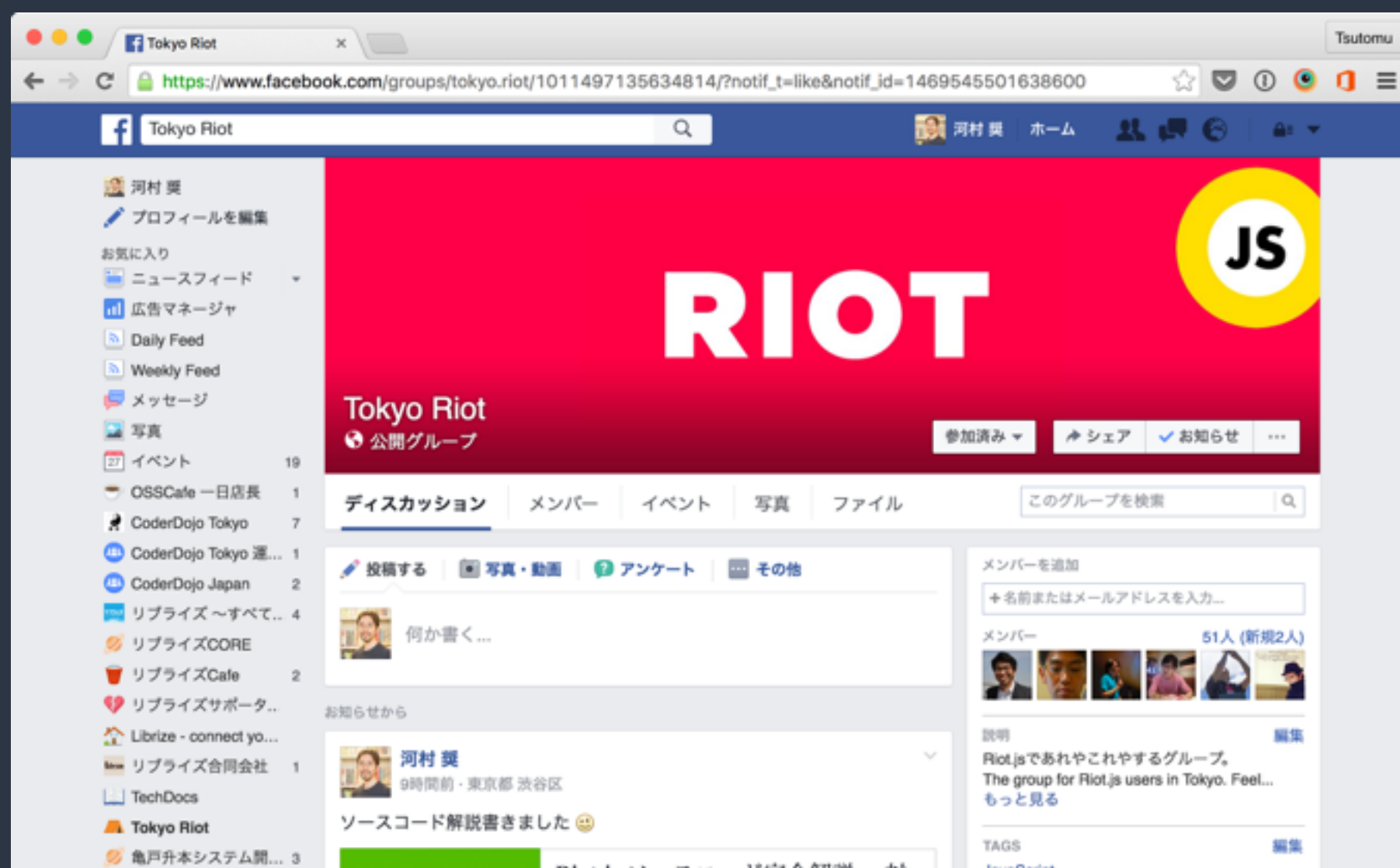
- ・モダンブラウザ以後のライブラリ
- ・ブラウザをある程度信頼 「ボールは友達」
- ・ブラウザに機能を足したい、じゃなくて
「ブラウザの機能で楽しみたい」

RIOT

Riot の話をするならこちら!

[fb.com/groups/tokyo.riot/](https://www.facebook.com/groups/tokyo.riot/)

gitter.im/riot/riot/



水・土曜は
下北沢で
店番してます

気軽に聞きに来て
ください!



OpenSource Cafe

下北沢 オープンソース・カフェ

by CogniTom