カジュアルなフロントエンドでこの先生きのこるには

Ayumu Sato

Oct 18, 2014 Frontrend In Kanazawa

photo https://www.flickr.com/photos/44534236@N00/11785665944

さとう　あゆむ

佐藤　歩

ハンドルネーム

@ahomu

年間維持費　￥8,480

http://aho.mu

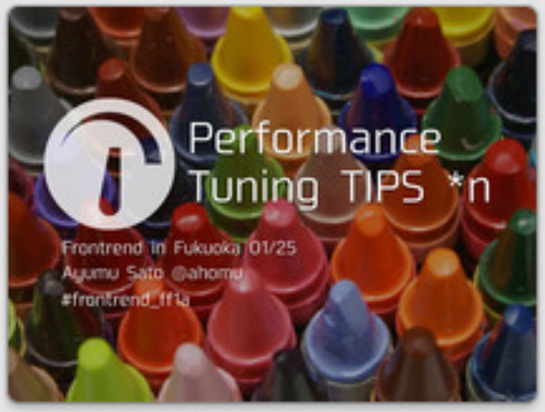
Talks by Ayumu Sato

Material Design with Polymer

Jul 28, 2014 by Ayumu Sato

フロントエンドのスキルマップと育成のはなし

Jun 21, 2014 by Ayumu Sato



Web Frontend Performance Tuning TIPS *n

Jan 25, 2014 by Ayumu Sato

俺的フロントレンド2013総括

Dec 18, 2013 by Ayumu Sato

Web Frontend Rendering Performance Knowledge & Tips

Nov 30, 2013 by Ayumu Sato

High Performance Web Frontend 2013 秋

Sep 14, 2013 by Ayumu Sato

jQuery to Backbone – into JavaScript architecture.

Feb 9, 2013 by Ayumu Sato

Stylusが目指すCSSプリプロセッサ

Jan 12, 2013 by Ayumu Sato

Profile

株式会社サイバーエージェント

アメーバ事業本部

Webフロントエンドエンジニア

Frontend + Node + Android + AWS

HTML5 Experts.jp 半幽霊部員

フロントエンドの中でも
テクニカルな面に
偏ったお話です

Agenda

フロントエンドの流行り廃りと距離感

現在のトレンドと選定の視点

フロントエンダの取捨選択

フロントエンドの流行り廃りと距離感

photo https://www.flickr.com/photos/kitkaphotogirl/1607221467

フロントエンド界隈のプロダクト

Grunt？ Gulp？

Yeoman？ broccoli？

styledocco？ KSS？ hologram？

Sass？ LESS？

Stylus？ Rework？

Backbone.js ?
KnockoutJS ?
Ember.js ?
AngularJS ?
React ?

3年くらい前から
急速に増加して
入れ替わり続けている

※個人の感想です

入れ替わりが
早すぎないか!?

GRUNT


BACKBONE.JS


ANGULARJS
by Google

枯れる前に次の技術へ
乗り換えていく

今までの経験を全て捨てるわけではないが
いくらかの学習コストは必要になる

なぜ？

「枯れた技術」のメリットをあまり活かそうとしないのか？

実行環境の多様化
デバイスの性能向上

高性能になれば表現が広がり、作る物が変われば作り方も変わる

流行り廃りと上手に付き合いたい

現在のトレンドと選定の視点

photo https://www.flickr.com/photos/gnilenkov/6234301745

1. タスクランナー
2. CSSプロセッサ
3. JavaScriptライブラリ
4. パッケージ管理
5. 標準仕様（HTML/CSS/JavaScript）

最も重要なのは
視点

どんなメリットを享受するために、どんなアプローチを選ぶのか
視点さえおさえれば、自分で判断できる

タスクランナー界隈

GRUNT

- エコシステムの発展具合
- 設定ファイルっぽい？
- スクリプトっぽい？

Getting Started Plugins Documentation

GRUNT

The JavaScript Task Runner

Grunt

http://gruntjs.com/

Ads by Bocoup.

Latest Version

- Stable: v0.4.5 (npm)
- Development: v0.4.6 (github)

Why use a task runner?

In one word: automation. The less work you have to do when performing repetitive tasks like minification, compilation, unit testing, linting, etc, the easier your job becomes. After you've configured it through a Gruntfile, a task runner can do most of that mundane work for you—and your team—with basically zero effort.

Why use Grunt?

The Grunt ecosystem is huge and it's growi every day. With literally hundreds of plugins choose from, you can use Grunt to automat just about anything with a minimum of effort someone hasn't already built what you need authoring and publishing your own Grunt pl to npm is a breeze. See how to get started.

Latest News

Grunt 0.4.5 released
May 12, 2014

Available Grunt plugins

```
grunt.initConfig({
  sass: {              // task
    dist: {            // target
      options: {  // options
        style: 'expanded'
      },
     files: {
        // dest: source
        'main.css': ‘main.scss’
      }
    }
  }
});
grunt.loadNpmTasks('grunt-contrib-sass');
```

```
grunt.initConfig
  sass:
    dist:
      options:
        style: 'expanded'
    files
       'main.css': ‘main.scss’

grunt.loadNpmTasks 'grunt-contrib-sass'
```

HOME DOCS CODE PLUGINS TWITTER

gulp.js

The streaming build system

Gulp

http://gulpjs.com/

Google Developers

View Source | Submit Feedback

Web **Starter Kit** Download Getting Started Docs GitHub

Web Starter Kit

Boilerplate & Tooling for Multi-Device Development

Download Kit (0.5.1)

Web Starter Kit

https://developers.google.com/web/starter-kit/index

```
var gulp = require('gulp');
var sass = require(‘gulp-ruby-sass')

gulp.task('sass', function () {
  gulp.src(‘./main.scss’)     // source
      .pipe(sass({            // task
        style : ‘expanded'    // options
      }))
      .pipe(gulp.dest(‘./‘));// dest
});
```

1タスク-n対象 の関係

```
grunt.initConfig({
  coffeehint: {
    src: [‘main.coffee’]
  },
  coffeescript: {
    files: {
      'main.js': ‘main.coffee’
    }
  },
  uglify: {
    files: {
      ‘main.min.js': ‘main.js’
    }
  }
});
```

1ファイルに対して適用されている
複数のタスクが散乱して流れを追いづらい

nソース-nタスク の関係

```
gulp.task(‘jsbuild', function () {
  gulp.src(‘./main.coffee’)
      .pipe(coffeelint())
      .pipe(coffee())
      .pipe(uglify())
      .pipe(gulp.dest(‘./‘));
});
```

指定したソースありきでタスクを
流れるように適用できて見通しが良い
(しかし、JavaScriptっぽい)

CSSプロセッサ界隈


less


Sass


LibSass


stylus

- 使いたい言語機能
- CSSからの移行・学習コスト
- プリプロセス or ポストプロセス

.sass
.less
.styl

プリ
プロセス

.css

ポスト
プロセス

.css

プリプロセス

Your laptop needs more Sass. Grab a set of Sass stickers now.

Sass

INSTALL LEARN SASS BLOG DOCUMENTATION GET INVOLVED LIBSASS

CSS with superpowers

SASS

http://sass-lang.com/

Sass is the most mature, stable, and powerful professional grade CSS extension language in the world.

Current Release: Selective Steve (3.4.5) Release Notes Fork on Github

CSS COMPATIBLE

Sass is completely compatible with all

FEATURE RICH

Sass boasts more features and abilities

MATURE

Sass has been actively supported for

Fork me on GitHub

WHAT? SASSC ABOUT

What?

LibSass

http://libsass.org/

Sass is a pre-processing language for CSS. It allows you to write cleaner stylesheets and makes collaboration on your CSS a breeze. There's a ton of information on Sass out there, so we won't repeat it all here. Just make sure to check out the Sass site for tutorials and examples.

Sass was originally written in Ruby. Ruby's great, but people started having a couple of issues. First, we want everyone to enjoy Sass, no matter what language they use. Why restrict everyone to using Ruby? In addition, Ruby can be kind of slow. Lowering compile time for users is important. Enter LibSass.

LibSass is a C/C++ port of the Sass engine. The point is to be simple, fast, and easy to integrate. Find out more about the project over at Github.

{less} Language Features Function Reference Usage About GitHub

Getting started

An overview of Less, how to download and use, examples and more.

less v1.7.5 has been released - **See what's new**

Less

http://lesscss.org/

Getting Started

Using Less

Command-line With Rhino

Client-side usage

Get Less.js

License FAQs

Getting Started

Less is a CSS pre-processor, meaning that it extends the CSS language, adding features that allow variables, mixins, functions and many other techniques that allow you to make CSS that is more maintainable, themable and extendable.

Less runs inside Node, in the browser and inside Rhino. There are also many 3rd party tools that allow you to compile your files and watch for changes.

For example:

```
@base: #f938ab;
```

Fork me on GitHub

Stylus

http://learnboost.github.io/stylus/

Expressive, dynamic, robust CSS

CSS needs a hero

```
body {
  font: 12px Helvetica, Arial, sans-serif;
}
a.button {
  -webkit-border-radius: 5px;
  -moz-border-radius: 5px;
  border-radius: 5px;
}
```

What if we could omit braces?

Readme.md

Download ZIP

rework

CSS manipulations built on `css`, allowing you to automate vendor prefixing, create your own properties, inline images, anything you can imagine!

Please refer to `css` for AST documentation and to report parser/stringifier issues.

Installation

```
$ npm install rework
```

rework

https://github.com/reworkcss/rework

Usage

```
var rework = require('rework');
var pluginA = require('pluginA');
var pluginB = require('pluginB');

rework('body { font-size: 12px; }', { source: 'source.css' })
  .use(pluginA)
  .use(pluginB)
  .toString({ sourcemap: true })
```

Readme.md

Styl

Work-in-progress CSS preprocessor. Spiritual successor of Stylus. Built on top of Rework.

Styl is basically an opinionated configuration of Rework. It does not aim for feature parity with Stylus.

If your application benefits from a runtime (conditionals, loops etc.), then Stylus is for you. If your application benefits from incredibly fast builds, simplicity, and the most transparent CSS preprocessor around, then Styl is for you.

Building Styl on top of Rework drastically reduces complexity. That's because Rework is comprised of multiple smaller pieces, plugins, and has no complex runtime. If you wish to include custom plugins, or configure Styl beyond its defaults, the interface is the same as Rework.

styl

https://github.com/visionmedia/styl

Installation

```
$ npm install -g styl
```

or with component:

```
$ component install component/styl
```

or with a script tag using ./styl.js

ポストプロセス

README.md

PostCSS build failing

PostCSS is a framework for CSS postprocessors, to modify CSS with JavaScript with full source map support.

It takes care of the most common CSS tool tasks:

1. parses CSS;
2. provides a usable JS API to edit CSS node trees;
3. dumps the modified node tree into a CSS string;
4. generates a source map (or modifies an pre-existing source map) containing your changes;

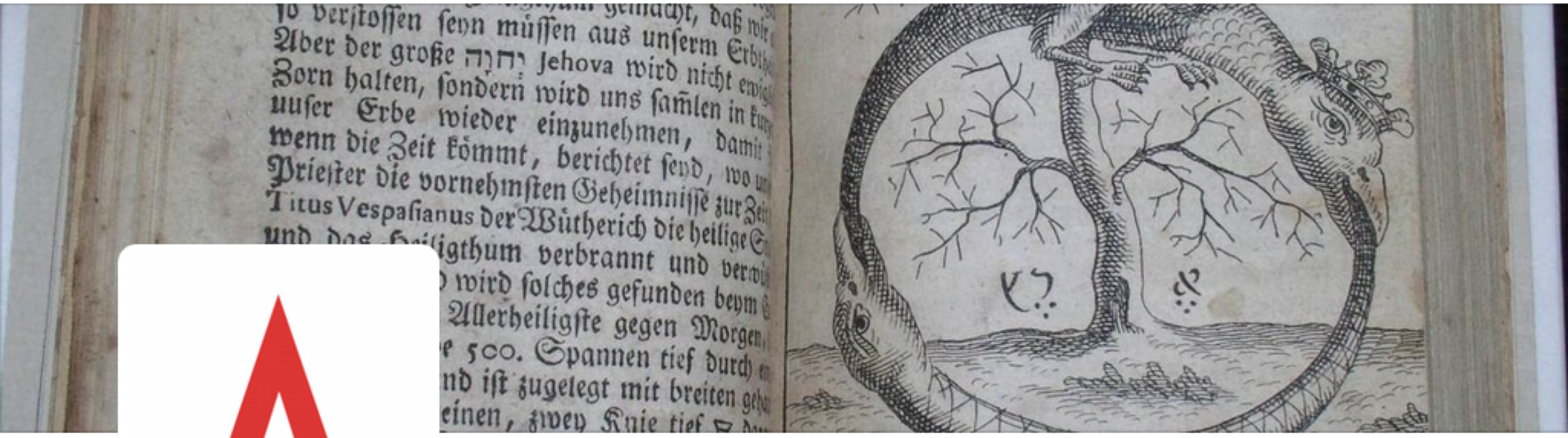
You can use this framework to write your own:

- CSS minifier or beautifier.
- CSS linter.
- Grunt plugin to generate sprites, include `data-uri` images or any other work.
- Text editor plugin to automate CSS routines.
- Command-line CSS tool.

PostCSS
https://github.com/postcss/postcss

Sponsored by Evil Martians.

Built with PostCSS



Autoprefixier

https://github.com/postcss/autoprefixer

```
/* Sass with Compass */
* {
  @include box-sizing(border-box);
}

/* CSS (compiled) */
* {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
```

```
/* CSS */
* {
  box-sizing: border-box;
}

/* CSS (autoprefixed) */
* {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
```

JavaScriptライブラリ界隈


ANGULARJS
by Google


BACKBONE.JS


React

連載： イベントレポート (23)

いまや最も優れたJavaScriptフレームワーク「AngularJSリファレンス」出版記念会

佐川 夫美雄

AngularJS

2014年9月22日

＿人人人人人人人人人人＿
＞ いまや最も優れた ＜
￣Y^Y^Y^Y^Y^Y^Y^Y^Y^Y^Y￣

AngularJSの機能や開発ライフサイクルなどがまとめられた「AngularJSリファレンス」が書籍がインプレスさから出版されました。これを記した著者でもある、池添 明宏氏(著)/金井 健一氏(著)/吉田 真也氏(著)/丸山 弘詩氏(編集)をお招きして「AngularJS」についてビール片手に軽く語っていただきました。

2014年9月18日（木）、場所はイベントの聖地21Cafeで開催されました。

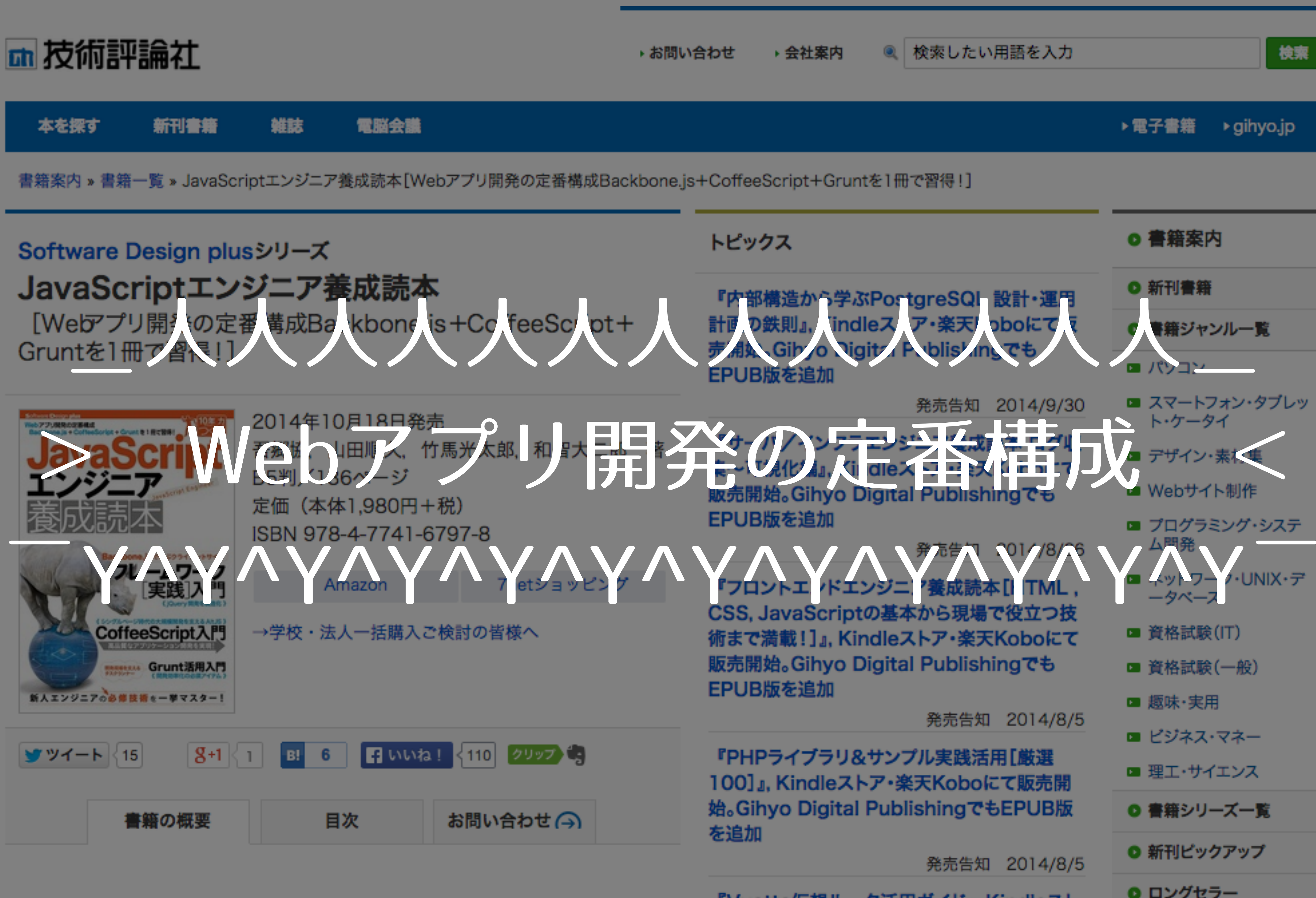
週間PVランキング > more

1 TypeScriptで既存資産を活かしたモダンWeb開発を実践！
2 JavaからHTML5へ。業務システム開発におけるWeb技術の変化と適応事
3 HTML5ハイブリッドアプリ開発、最新動向はやわかり！
4 Bootstrap3超速レビュー！刷新されたグリッドシステムを理解しよう！
5 初心者でも分かる、WebGLプログラミング<three.js最初の一歩>

> more

TypeScriptで既存資産を活かしたモダンWeb開発を実践！
Webの未来を議論！「Extensible Web Summit Berlin」イベントレポート
開発者によるWeb標準化を可能とする「Extensible Web」とは？
いまや最も優れたJavaScriptフレームワーク「AngularJSリファレンス」出版記念会
サンフランシスコから速報！TechCrunch

＿人人人人人人人人人人人人人＿
＞ Webアプリ開発の定番構成 ＜
￣Y^Y^Y^Y^Y^Y^Y^Y^Y^Y^Y^Y^Y￣



- 解決しようとしている範囲
- アーキテクチャモデル
- DOMテンプレーティング
  - Data Binding
  - Virtual DOM
- 宣言的 or 命令的

BACKBONE.JS

Backbone.js gives structure to web applications by providing **models** with key-value binding and custom events, **collections** with a rich API of enumerable functions, **views** with declarative event handling, and connects it all to your existing API over a RESTful JSON interface.

The project is hosted on GitHub, and the annotated source code is available, as well as an online test suite, an example application, a list of tutorials and a long list of real-world projects that use Backbone. Backbone is available for use under the MIT software license.

You can report bugs and discuss features on the GitHub issues page, on Freenode IRC in the `#documentcloud` channel, post questions to the Google Group, add pages to the wiki or send tweets to @documentcloud.

Backbone is an open-source component of DocumentCloud.

Downloads & Dependencies (Right-click, and use "Save As")

Development Version (1.1.2) *60kb, Full source, tons of comments*

Window size: **1024** x **742**
Viewport size: **1024** x **626**

Backbone.JS

http://backbonejs.org/

```
<div>
<button class="js-ouch"><%= foo %></button>
</div>

Backbone.View.extend({
  initialize: function() { this.render(); },
  events: {
    'click .js-ouch': 'onOuch'
  },
  onOuch: function() {
    alert('ouch!!');
  },
  render: function() {
    var newHtml = ejsTmpl({foo: 'bar'});
    return this.$el.html(newHtml);
  }
});
```

ANGULARJS | Home | Learn | Develop | Discuss | Search

HTML enhanced for web apps!

AngularJS

https://angularjs.org/

View on GitHub

Download (1.2.26 / 1.3.0-rc.5)

Follow +AngularJS on g+1 +88410

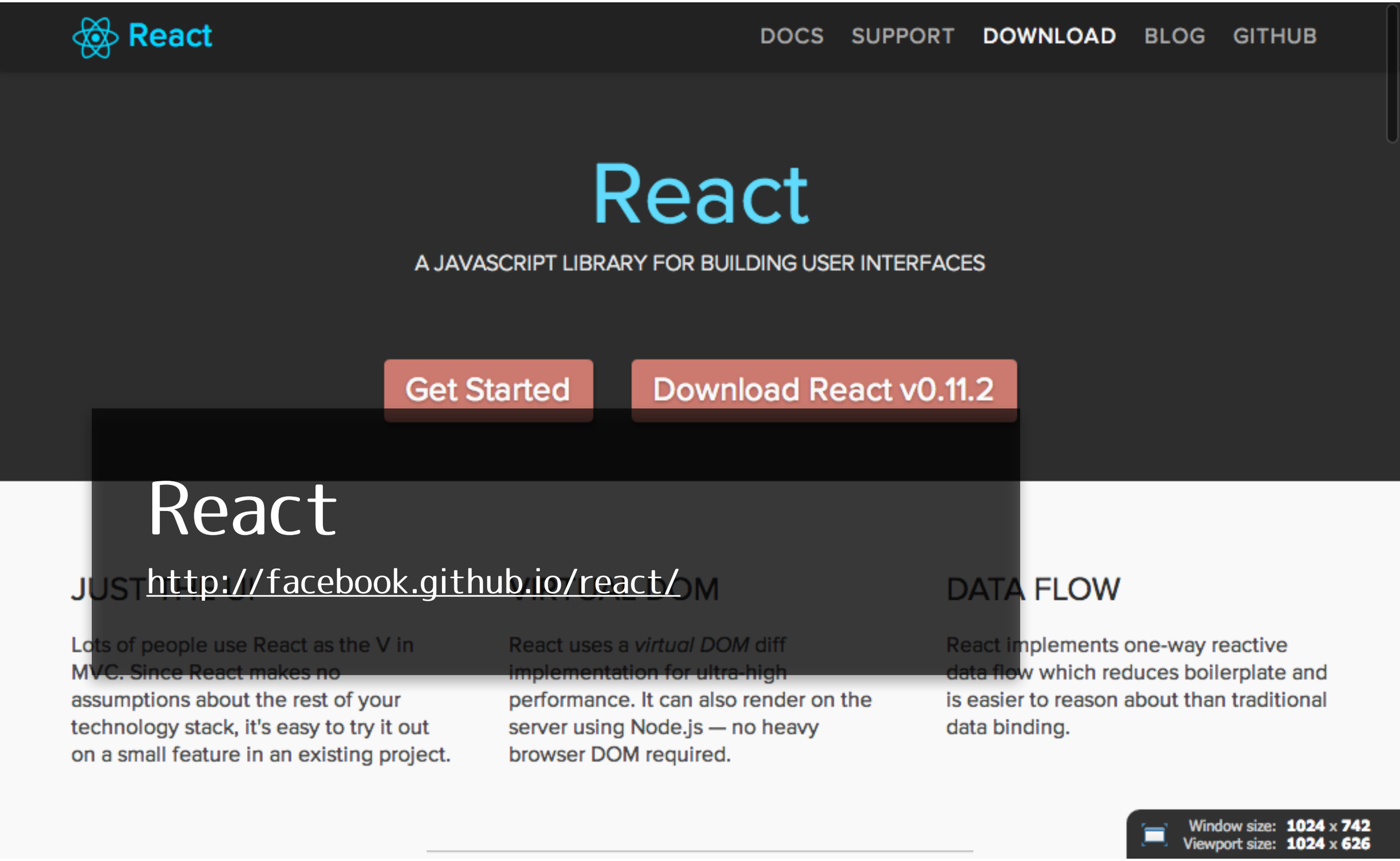
Follow @angularjs 49.1K followers

Tweet 4,933

Learn Angular in your browser for free!

Window size: **1024** x **742**
Viewport size: **1024** x **626**

```
<div ng-contoller="OuchCtrl">
  <button ng-click="onOuch()">{{foo}}</button>
</div>

angular
  .module('app')
  .controller('OuchCtrl', function($scope) {
    $scope.foo = 'bar';

    $scope.onOuch = function() {
      alert('ouch!!');
    };
});
```


React
DOCS SUPPORT DOWNLOAD BLOG GITHUB
React
A JAVASCRIPT LIBRARY FOR BUILDING USER INTERFACES
Get Started
Download React v0.11.2
React
http://facebook.github.io/react/
DATA FLOW
Lots of people use React as the V in MVC. Since React makes no assumptions about the rest of your technology stack, it's easy to try it out on a small feature in an existing project.
React uses a virtual DOM diff implementation for ultra-high performance. It can also render on the server using Node.js — no heavy browser DOM required.
React implements one-way reactive data flow which reduces boilerplate and is easier to reason about than traditional data binding.
Window size: 1024 x 742
Viewport size: 1024 x 626

```
<script type="text/jsx">
/** @jsx React.DOM */
var OuchComponent = React.createClass({
  onOuch: function() {
    alert('ouch!!');
  },
  getInitialState: function() {
    return {foo: 'bar'};
  },
  render: function() {
    return (
      <div>
        <button onClick={this.onOuch}>
        {this.state.foo}</button>
      </div>
    );
  }
});
</script>
```

```jsx
/** @jsx React.DOM */
var HelloMessage = React.createClass({
  render: function() {
    return <div>Hello {this.props.name}</div>;
  }
});
```

```jsx
var HelloMessage = React.createClass({
  displayName: 'HelloMessage',
  render      : function() {
    return React.DOM.div(null,
                         "Hello ",
                         this.props.name);
  }
});
```

パッケージ管理＆モジュールシステム界隈

webpack

MODULE BUNDLER

- 色々あるけど、役割の違うツールが多い
- パッケージのインストール補助？
- ランタイム or ビルド時の依存解決？
  - 標準仕様との互換性は？
- それとも両方の役割をできるもの？

パッケージ管理

HOME
API
BLOG
NODE.JS
JOBS

WHO'S HIRING
JUT
+ 13 MORE...

npm Enterprise
Try the on-premises solution for private npm

Search Packages

Create Account | Login

Node Packaged Modules

Total Packages: 99 381

22 009 070	downloads in the last day
133 199 273	downloads in the last week
550 799 779	downloads in the last month

Patches welcome!

Any package can be installed by using `npm install`.

Add your programs to this index by using `npm publish`.

Recently Updated

- 1m startserver
- 1m sa
- 2m p2r
- 8m generator-angular-require
- 8m generator-derby
- 8m avoscloud-sign
- 8m linda-socket.io
- 8m validator-extended
- 10m ?vat-model

Most Depended Upon

- 7258 underscore
- 6750 async
- 5876 request
- 5374 lodash
- 3795 commander
- 3696 express
- 2740 optimist
- 2708 coffee-script
- 2703 colors

npm

https://www.npmjs.org/

Docs Search packages Stats

Bower

A package manager for the web

Bower

http://bower.io/

Home
- Install Bower
- Getting started
- @bower

Creating packages

API

Configuration

Tools

About

Web sites are made of lots of things — frameworks, libraries, assets, utilities, and rainbows. Bower manages all these things for you.

Bower works by fetching and installing packages from all over, taking care of hunting, finding, downloading, and saving the stuff you're looking for. Bower keeps track of these packages in a manifest file, `bower.json`. How you use packages is up to you. Bower provides hooks to facilitate using packages in your tools and workflows.

Bower is optimized for the front-end. Bower uses a flat dependency tree, requiring only one version for each package, reducing page load to a minimum.

モジュールシステム

A JAVASCRIPT
MODULE LOADER


```
/* ---

RequireJS is a JavaScript file and module
loader. It is optimized for in-browser use, but
it can be used in other JavaScript environments,
like Rhino and Node. Using a modular script
loader like RequireJS will improve the speed and
quality of your code.

IE 6+ ..... compatible ✔
Firefox 2+ ..... compatible ✔
Safari 3.2+ .... compatible ✔
Chrome 3+ ...... compatible ✔
Opera 10+ ...... compatible ✔

Get started then check out the API.

--- */
```

RequireJS

http://requirejs.org/

Latest Release: 2.1.15

Open source: new BSD or MIT licensed

```
# html
<script src="require.js" data-
main="main.js" async></script>

# main.js
define([‘module’], function(module) {
  alert(module.foo); // ‘bar’
});

# module.js
define(function() {
  return {foo: ‘bar’}
});
```

Browserify lets you require('modules') in the browser by bundling up all of your dependencies.

INSTALL | DOCUMENTATION | HELP + ARTICLES | SWEET DEMOS

browserify

http://browserify.org/

Install

USE BROWSERIFY FROM THE COMMAND LINE

First install node, which ships with npm. Then do:

```
#index.js

// from node_modules
var htmlparser = require('htmlparser');
// from local
var module = require('./module');
// from bower ( recommend: debowerify )
var jquery = require('./bower_components/jquery/
dist/jquery.js')
```

- -

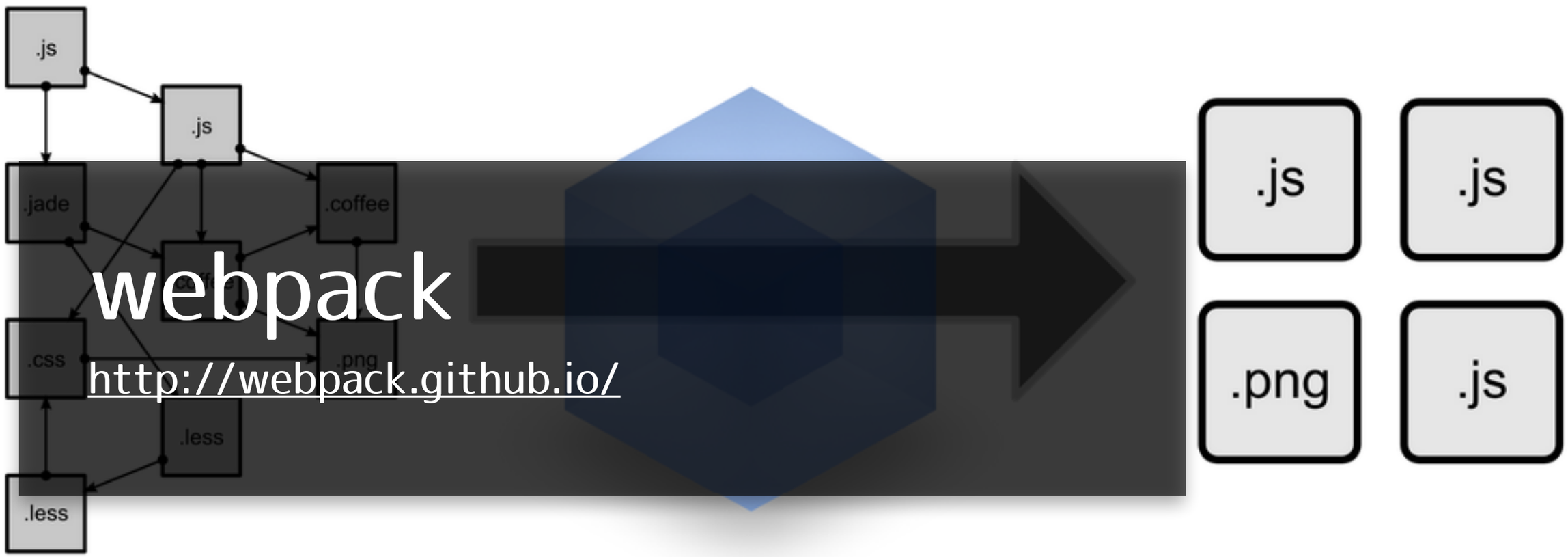
```
#module.js

module.exports = {
  foo: 'bar'
};
```

Fork me on GitHub

webpack

MODULE BUNDLER

Getting started by reading the **documentation**, which also contains an **introduction** and a **tutorial**.



modules with dependencies

webpack

MODULE BUNDLER

static assets

```
var path = require("path");
var CommonsChunkPlugin = require("../../lib/
optimize/CommonsChunkPlugin");
module.exports = {
   entry: {
      pageA: "./pageA",
      pageB: "./pageB"
   },
   output: {
      path: path.join(__dirname, "js"),
      filename: "[name].bundle.js",
      chunkFilename: "[id].chunk.js"
   },
   plugins: [
      new CommonsChunkPlugin("commons.js")
   ]
}
```

標準仕様の界隈

ここだけ

普通に時勢の話です


HTML


CSS

W3C Editor's Draft

Select text and
file a bug

Add developer-view styles

HTML 5.1 Nightly

A vocabulary and associated APIs for HTML and XHTML

Editor's Draft 11 October 2014

Latest Published Version:
http://www.w3.org/TR/html51/
Latest Version of HTML:
http://www.w3.org/TR/html/
Latest Editor's Draft:
http://www.w3.org/html/wg/drafts/html/master/
Previous Versions:
http://www.w3.org/TR/2014/WD-html51-20140204/
Editors:
WHATWG:
Ian Hickson, Google, Inc.
W3C:
Robin Berjon, W3C
Steve Faulkner, The Paciello Group
Travis Leithead, Microsoft
Erika Doyle Navara, Microsoft
Edward O'Connor, Apple Inc.
Silvia Pfeiffer
For the `img` and `picture` elements:
Tab Atkins (Google)
Simon Pieters (Opera Software)
Yoav Weiss

HTML 5.1 Nightly
http://www.w3.org/html/wg/drafts/html/master/

22 July 2014 by Andreas Bovens in Articles. Translated by Masataka Yakura.

Tags: html media media-queries picture rwd

Translations: Japanese English

Edit this article on GitHub

レスポンシブ・イメージ：ユースケースと入門用のコードサンプル

はじめに

ついに、本物のレスポンシブイメージが Web で使えるようになりました。HTML だけで完結し、面倒なハックはありません。新しい `<picture>` 要素と、`<img>` に追加された新しい属性が Chrome 38 から使えます（なので Opera でも使えます）。他のブラウザは、Firefox のナイトリービルドで実装されており、WebKit では実装中です。

`<picture>` はいくつかのユースケースに対して作られた要素のため、コードがごちゃごちゃすることがあります。提供したいレスポンシブイメージの書き方がどれか調べられるように、この記事ではそれぞれのケースに対応するサンプルコードを紹介します。

レスポンシブ・イメージ：
ユースケースと入門用のコードサンプル
https://dev.opera.com/articles/ja/responsive-images/

4つの質問

レスポンシブイメージを使う前には、次の問いに答えるようにしましょう。

- サイトのデザインが変化するのに応じて、画像の**サイズ**も変化させたい

Dev.Opera Blog Articles Extensions TV Search Dev.Opera

dialog element: Modals made easy

By Eiji Kitamura

Published: September 23, 2013
Updated: September 23, 2013
Comments: 0

Chrome Canary has landed support for the dialog element behind a flag. The `dialog` element can be used for popups in a web page.

- `show()`: Open dialog.
- `close()`: Close dialog. Takes an optional argument which if present sets `returnValue` is set to.
- `showModal()`: Open modal dialog.
- `::backdrop`: Pseudo-element to style background behind a modal dialog.
- `close` event: Fired when a dialog is closed.

Update on Dec 16th 2013

The `dialog` element now supports:

dialog element: Modals made easy

http://updates.html5rocks.com/2013/09/dialog-element-Modals-made-easy

HOME TUTORIALS UPDATES CONTRIBUTE SLIDES RESOURCES

W3C Editor's Draft

Selectors Level 4

Editor's Draft, 7 October 2014

This version:

http://dev.w3.org/csswg/selectors

Latest version:

http://www.w3.org/TR/selectors/

Previous Versions:

http://www.w3.org/TR/2013/WD-selectors4-20130502/

http://www.w3.org/TR/2012/WD-selectors4-20120823/

http://www.w3.org/TR/2011/WD-selectors4-20110929/

Feedback:

www-style@w3.org with subject line "`[selectors] … message topic …`" (archives)

Editors:

Elika J. Etemad (Invited Expert)

Tab Atkins Jr. (Google)

Former Editors:

Tantek Çelik

Daniel Glazman

Ian Hickson

Peter Linss

Selectors Level 4

http://dev.w3.org/csswg/selectors4/

CSS SPECIFICATIONS

http://www.w3.org/Style/CSS/current-work

Refining	Current	Upcoming	Notes
CSS Animations	WD	WD	
CSS Counter Styles Level 3	LC	CR	
CSS Text Level 3	LC	CR	
CSS Fragmentation Level 3	WD	WD	
CSS Transforms	WD	WD	
CSS Transitions	WD	LC	
Cascading Variables	WD	LC	
Compositing and Blending	LC	CR	
CSS Syntax Level 3	CR	PR	

Revising	Current	Upcoming	Notes
CSS Box Alignment Module Level 3	WD	WD	
CSS Grid Layout	WD	WD	
CSS Page Media Level 3	WD	LC	
CSS Basic User Interface Level 3	CR	LC	
CSSOM View	WD	WD	
Selectors Level 4	WD	WD	

Exploring	Current	Upcoming	Notes
CSS Backgrounds and Borders Level 4		WD	
CSS Device Adaptation	WD	WD	
CSS Exclusions	WD	WD	
Filter Effects	WD	WD	
CSS Generated Content for Paged Media	WD	WD	
CSS Page Floats		WD	

Views: desktop mobile print

STANDARDS | PARTICIPATE | MEMBERSHIP | ABOUT W3C

Google™

W3C » Standards » All Standards and Drafts

Skip

STANDARDS

- Web Design and Applications
- Web Architecture
- Semantic Web
- XML Technology
- Web of Services
- Web of Devices
- Browsers and Authoring Tools
- All Standards and Drafts
- About W3C Standards

ALL STANDARDS AND DRAFTS

Select All sorted by Technology Show View

Show details Hide details

- Accessibility (All)
- Accessible Rich Internet Applications (WAI-ARIA)
- Audio
- Authoring Tool Accessibility Guidelines (ATAG)
- Best Practices for Authoring HTML
- CC/PP
- Content Transformation
- CSS
- CSS Mobile
- CSV on the Web
- DCCI
- Declarative Web Applications
- Device Description Repository
- Device Independence Authoring
- Digital Publishing
- DOM
- DOM events

All Standards and Drafts

http://www.w3.org/TR/

WebComponents.org

a place to discuss and evolve web component best-practices

WHAT?

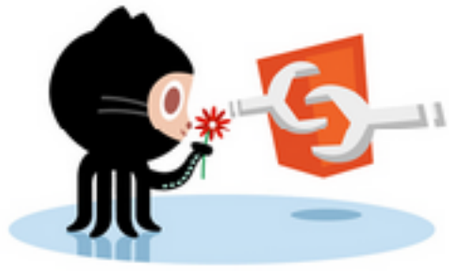
WebComponents.org is where pioneers and community-members of the Web Components ecosystem (like *Polymer*, *X-tags*, and other interested parties) document web components best practices so that others can follow the same path instead of needlessly striking out on their own.

SPECS

WEB COMPONENTS

This document is a non-normative reference, which provides an overview of Web Components. It summarizes the normative information in the respective specifications in easy-to-

ARTICLES



HOW GITHUB IS USING WEB COMPONENTS IN PRODUCTION

More and more people have been using Web Components. Some just

Custom Elements `Document#registerElement()`

HTML Imports `<link rel="import" href="…">`

Shadow DOM `Element#createShadowRoot()`

Templates `<template></template>`


ecma
INTERNATIONAL

[[harmony:specification_drafts]]

ES Wiki

Show pagesource | Old revisions | Recent changes | Search

Trace: » specification_drafts

Draft Specification for ES.next (Ecma-262 Edition 6)

Table of Contents ▲

This page contains a historical record of working draft of the ES.next specification prepared by the project editor.

Errors in the current draft should be reported as bugs at bugs.ecmascript.org. Report the version you are reading, and an appropriate "component" (editorial issues, technical issues, etc.). Such bug reports are particularly useful for specific bugs such as misspellings or errors in algorithms.

An unofficial HTML version of the Edition 6 working draft is usually available at http://people.mozilla.org/~jorendorff/es6-draft.html]. Note that this HTML version may not always be up to date with the current working draft below.

Some Guidelines for Reading these Drafts

For now, all draft found here are snapshots taken of the project editor's working draft. Snapshots are taken at fairly arbitrary points in time. The drafts are still far from being complete, often have incomplete treatment of new features, and sometimes are captured in the middle of some major refactoring. Many features that are intended for the next edition may not yet be included in any particular draft. Incomplete material and undecided issues are often marked with editorial notes within the document.

ES6 (ES.next, harmony)

http://wiki.ecmascript.org/doku.php?id=harmony:specification_drafts

It is reasonable to assume that the editor is aware of the major inconsistencies or missing material, but you can't over report. So if you see something that you think is important, capture file a bug.

The rationale for design decisions, are not generally part of the specification. You can find background information on this wiki and in the archives of the es-discuss mailing list. If you want to discuss why certain decisions were made or want to argue for an alternative then you should post messages about specific issues to es-discuss.

The primary target audience of this specification is implementors who must create highly interoperable ECMAScript implementations. In one sense, the specification is all about the edge cases that implementors must get exactly right in order to be completely interoperable. Next comes the authors of interoperability and conformance test suites, such as test262. They need to write tests that help implementors conform to the specification. Third in priority, are sophisticated authors and educator. These are the people who will teach the language to the world, and like implementors, they need to know (or at least lookup) edge case behavior.

This is not intended as a document for everyday ECMAScript coders (although some will use it). It is also not a tutorial on either language design in general, language theory, or the design process behind its creation. The spec. also isn't particularly targeted towards academics. We certainly want to have a sound language and where necessary TC39 members may prove characteristics of certain features. But such material is not necessarily included in the specification.

We generally within the specification try to minimize tutorial material (eg examples) and redundancy between normative prose descriptions and normative

README.md

build passing

What is Traceur?

Traceur is a JavaScript.next-to-JavaScript-of-today compiler that allows you to use features from the future today. Traceur's goal is to inform the design of new JavaScript features which are only valuable if they allow you to write better code. Traceur allows you to try out new and proposed language [illegible] forming the standards process.

traceur-compiler

https://github.com/google/traceur-compiler

JavaScript's evolution needs your input. Try out the new language features. Tell us how they work for you and what's still causing you to use more boilerplate and "design patterns" than you prefer.

What now? What can Traceur do for me?

Read the Getting Started page to get up and running. You can use some language features right now and even try it out in your browser here. Just type in some code and see what Traceur produces. For an idea of what is available and what we have in the pipeline, see the Language Features page.

README.md

travis failing | code climate 4.0 | coverage 100% | dependencies up-to-date

6to5

https://github.com/sebmck/6to5

6to5 turns ES6 code into vanilla ES5, so you can use ES6 features **today.**

- **Readable** generated code - designed so that generated code is as similar as possible.
- **Extensible** - with a large range of plugins and **browser support**.
- **Lossless** - **source map support** so you can debug your compiled code with ease.
- **Compact** - maps directly to the equivalent ES5 with **no runtime**.

Installation

It's as easy as:

これらの賞味期限が短い
トレンドとの向き合い方

紹介した内容は3年も持たないようなトレンドがほとんど
どうすべきか？

フロントエンダの取捨選択

photo https://www.flickr.com/photos/jmv/2857858757

技術は技術でしかなく
それを以てなにをするか

トレンドの取捨選択

ロールモデルに合った
トレンドを学び続ける

得意のアニメーションで
ゲームを作るひと？

デザインの立案から
実装までワンストップで
行えるひと？

全体のアーキテクトを
踏まえてフロント領域を
管理できるひと？

アクセシビリティ実装の専門家？

技術・デザイン・ビジネスの間を取り持ってディレクションできるひと？

学習すべきは
技術トレンドだけでは
ないのかもしれない

無意識にしている判断を
明確化するだけで
普段の学びが変わる

まとめ

- フロントの流行は3年くらい前から
  急速に変化するようになっている

- 自らの視点と判断を持ってトレンドに
  流されてしまわないようにする

- ロールモデルから逆算して自分にとって
  本当に必要な技術トレンドを学ぶ

10年先のため
今身につけたいこと

自分のロールモデルから逆算して学び続けること

10年先生きのこる！

ロールモデルも10年の間にアップデートし続けよう！

Questions?

- http://aho.mu
- @ahomu
- github.com/ahomu

photo https://www.flickr.com/photos/gnilenkov/6109406698