

CREATIVE ACADEMY

WEEK ONE

YUYA SAITO



CyberAgent

AN INTRO TO

HTML, CSS

& JAVASCRIPT



CyberAgent

SET UP



Welcome to JS Bin – JS Bin

jsbin.com/welcome/1/edit

JS Bin Add library Share HTML CSS JavaScript Console Output studiomohawk Help

Output

Run with JS Auto-run JS

Welcome to JS Bin!

Hello, I'm *Dave the JS Bin bot*. JS Bin is a JavaScript, HTML and CSS playground. If you want to experiment, learn or help your friends with a problem, then JS Bin is for you. Have a play and build something cool — and don't forget to follow me on Twitter! Oh, and please don't click on, I get rather dizzy...

Here's just a few of the things I can do:

Saving

As you type, HTML, CSS & JavaScript is automatically rendered live in this window. I'll save the bin for you too as you type.

History

If you're registered and logged in, I'll remember all your bins and give you a nice complete list with

Sharing

You can share bins, but the output is automatically refreshed as you edit this page.



```

<!DOCTYPE html>
<html>
<head>
<script src="http://jashkenas.github.com/coffee-script/extras/coffee-script.js"></script>
<meta charset=utf-8 />
<title>Welcome to JS Bin</title>
<meta name="viewport" content="width=device-width">
<link href='http://fonts.googleapis.com/css?family=Doppio+One' rel='stylesheet' type='text/css'>
<script class="jsbin" src="http://ajax.googleapis.com/ajax/libs/jquery/1/jquery.min.js"></script>
</head>

<body>
<div class="wrapper">


<div id="welcome">
<h1>Welcome to JS Bin!</h1>
<p class="bubble"><strong>Hello</strong>, I'm <em><strong>Dave</strong> the <strong>JS Bin</strong> bot</em>. <br/>JS Bin is a JavaScript, HTML and CSS playground. If you want to experiment, learn or help your friends with a problem, then JS Bin is for you.<br/>Have a play and build something cool &mdash; and don't forget to <a target="_blank" href="http://twitter.com/js_bin">follow me on Twitter</a>!<small id="message">Oh, and please don't click on, I get rather dizzy...</small></p>
<h2>Here's just a few of the things I can do:</h2>
</div>

<div id="features">
<ol>
<li><h2>Saving</h2>As you type, HTML, CSS & JavaScript is automatically rendered live in this window. I'll save the bin for you too as you type.</li>
<li><h2>History</h2>If you're <a class="open"

```

<http://jsbin.com>

```

<h2>Get meddling!</h2>
<p>Open the <a class="open" href="http://jsbin.com/welcome/edit?css, live#css">CSS</a> panel and try changing things about! You'll see this window update live (and get your own unique copy of this bin right away).</p>
<p>You won't see this message again by default, but if you do want a reminder, you can either use the <a class="open" href="#helpmenu">help</a> or visit <a href="http://jsbin.com/welcome/edit?>
```

Open the CSS panel and try changing things about! You'll see this window update live (and get your own unique copy of this bin right away).

You won't see this message again by

JS Bin is an open source project available on github with instructions on how to host it yourself.

GitHub · Social Coding

github.com

github

Signup and Pricing Explore GitHub Features Blog Sign in

2,015,968 people hosting over 3,535,884 repositories

jQuery, reddit, Sparkle, curl, Ruby on Rails, node.js, ClickToFlash, Erlang/OTP, CakePHP, Redis, and many more

Find any repository

facebook.  Microsoft® vmware®  redhat.  mozilla

git /'git/

Git is an extremely fast, efficient, distributed version control system ideal for the collaborative development of software.

git·hub /'git,həb/

GitHub is the best way to collaborate with others. Fork, send pull requests and manage all your **public** and **private** git repositories.

Plans, Pricing and Signup

Unlimited public repositories are free!

Free public repositories, collaborator management, issue tracking, wikis, downloads, code review, graphs and much more...

Team management

30 seconds to give people access to code. No SSH key required. Activity

Code review

Comment on changes, track issues, compare branches, send pull requests

Reliable code hosting

We spend all day and night making sure your repositories are **secure**.

Open source collaboration

Participate in the most important open source community in the world today—

<https://github.com>

用語

HTML



CyberAgent

ELEMENTS 要素

body, h1, p, div, strong, em, span

ATTRIBUTES 属性

href="http://cyberagent.co.jp" title="CA"

TAG タグ

```
<div id="cyberagent" class="awesome"></div>
```



CyberAgent

用語

CSS



CyberAgent

SELECTOR セレクタ

```
h1 {  
font: bold 16px/24px Arial, sans-serif; }
```

PROPERTY プロパティ

```
h1 {  
  font: bold 16px/24px Arial, sans-serif; }
```

VALUE 值

```
h1 {  
font: bold 16px/24px Arial, sans-serif; }
```



CyberAgent

DECLARATIONS 宣言

```
h1 {  
font: bold 16px/24px Arial, sans-serif; }
```



CyberAgent

RULE SET ルールセット

```
h1 {  
font: bold 16px/24px Arial, sans-serif; }
```

クロスブラウザ対応

RESETTING STYLE

RESET.CSSの歴史



CyberAgent

2004

undohtml.css by Tantek Çelik

(<http://jsbin.com/ukocac/1/edit>)

hard reset by Andrew Krespanis

(<http://jsbin.com/ipohur/1/edit>)

"no assumptions" reset by Steve Rider

(<http://jsbin.com/ovukej/1/edit>)

Global White Space Reset by Andrew Krespanis

(<http://jsbin.com/idirok/1/edit>)



CyberAgent

2005

initial.css by Faruk Ateş

(<http://jsbin.com/umagol/1/edit>)

2005

YUI Reset by Nate Koechley

(<http://jsbin.com/iwogiz/1/edit>)



CyberAgent

2007

CSS Reset (rev.1) by Eric Meyer

(<http://jsbin.com/ebibag/1/edit>)

LATEST

CSS Reset 2.0 (20110126) by Eric Meyer

(<http://jsbin.com/ozoveg/1/edit>)



CyberAgent

HTML5

html5doctor.com Reset by Richard Clark
(<http://jsbin.com/akezoq/1/edit>)



CyberAgent

GOAL

Reset.cssは
ブラウザが定義するスタイルを上書きし、
スタイルされていない状態にすることを
目的にしている。

QUESTION

モダンブラウザでスタイルをリセットする必要があるか？



CyberAgent

ANSWER

YES AND NO



CyberAgent

NORMALIZE STYLE

NORMALIZE.CSS

NORMALIZE.CSS

“[...] a customizable CSS file that makes browsers render all elements more **consistently** and in line with **modern standards**. We researched the differences between default browser styles in order to **precisely target only the styles that need normalizing.**”

- Nicolas Gallagher



CyberAgent

NORMALIZE.CSS

- ▶ 有用なデフォルトスタイルは残す
- ▶ 幅広い範囲のHTML要素のスタイルを標準化
- ▶ バグ / ブラウザ間のスタイルの揺れを修正
- ▶ ユーザビリティの向上

A screenshot of a web browser window displaying the [normalize.css](http://necolas.github.com/normalize.css/) homepage. The title bar reads "Normalize.css: Make browsers render all elements more consistently." The address bar shows the URL "necolas.github.com/normalize.css/". The main content features a large green **normalize.css** logo, followed by the subtitle "A modern, HTML5-ready alternative to CSS resets". Below this is a detailed paragraph explaining what Normalize.css does. A section titled "What does it do?" lists six bullet points. At the bottom are two buttons: a green "Download v1.0.0" button and a grey "View on GitHub" button.

Normalize.css: Make browsers render all elements more consistently.

necolas.github.com/normalize.css/ Reader

normalize.css

A modern, HTML5-ready alternative to CSS resets

Normalize.css is a customisable CSS file that makes browsers render all elements more consistently and in line with modern standards. We researched the differences between default browser styles in order to precisely target only the styles that need normalizing.

What does it do?

- **Preserves useful defaults**, unlike many CSS resets.
- **Normalizes styles** for a wide range of HTML elements.
- **Corrects bugs** and common browser inconsistencies.
- **Improves usability** with subtle improvements.
- **Explains what code does** using detailed comments.

[Download v1.0.0](#) [View on GitHub](#)

<http://necolas.github.com/normalize.css/>

How do I use it?

Normalize.css is intended to be used as an alternative to CSS resets. It's suggested that you read the `normalize.css` file and consider customising it to meet your needs. Alternatively, include the file in your project and override the defaults later in your CSS.

SELECTOR



CyberAgent

TYPE SELECTOR

< p > . . . </ p >

p { . . . }

CLASS SELECTOR

```
<div class="awesome"> . . . </div>
```

```
.awesome { . . . }
```

ID SELECTOR

```
<div id="cyberagent">...</div>
```

```
#cyberagent { ... }
```

COMBINE SELECTOR

```
ul#social li { padding: 0 3px; }
ul#social li a {
  height: 17px;
  width: 16px;
}
ul#social li.tumblr a {
  background: url('tumblr.png') 0 0
    no-repeat;
}
```

SPECIFICITY

Specificity(詳細度)は
同じ要素に対して複数の宣言ブロックがある場合に
どちらを優先するかを確定するための仕様。



CyberAgent

QUESTION

```
<p id="myDiv" class="myClass">  
このパラグラフは何色？</p>
```

```
#myDiv { color: red; }  
.myClass {color: blue;}  
p {color: yellow;}
```



CyberAgent

CSS Selectors *more to come (suggest?)

[Tweet](#) - 637

- first-of-type
- first-child
- descendant
- child
- nth-child
- adjacent sibling
- general sibling
- first-line
- nth-of-type
- last-of-type

Currently applied CSS:

none

```
<div id="target">
  <h2>Where the Buggalo Roam</h2>
  <p>Alright, let's mafia things up a bit. Joey,
  burn down the ship. Clamps, burn down the
  crew. No! Don't jump! Dr. Zoidberg, that
  doesn't make sense.</p>
<div>
  <p>Does anybody else feel jealous and
  aroused and worried?</p>
</div>
<h2>The Mutants Are Revolting</h2>
<p>But, okay! I just want to talk. It has
  nothing to do with mating. Fry, that doesn't
  make sense.</p>
<h2>Space Pilot 3000</h2>
```

<http://twostepmedia.co.uk/cssselectors/>

<h1>CSS SpecifIshity</h1>			
with Plankton, Fish and Sharks			
universal selector 0-0-0	1 element 0-0-1	2 elements 0-0-2	14 elements 0-0-14
1 class 0-1-0	1 universal selector 1 class 0-1-0	1 universal selector 1 attribute selector 0-1-0	1 pseudo-class 0-1-0
li.myClass	li[attr]	li:nth-of-type(3n)-li	form input[type=email]

<http://www.standardista.com/css3/css-specificity/>

1 element 1 class 1 pseudo-class 0-2-1	1 element 1 class 1 attribute 0-2-1	10 class/attribute/ pseudo-classes 0-10-0	1 ID Selector 1-0-0
#myDiv li.class a[href]	#divitis #myDiv a	style=""	!important

BOX MODEL



CyberAgent

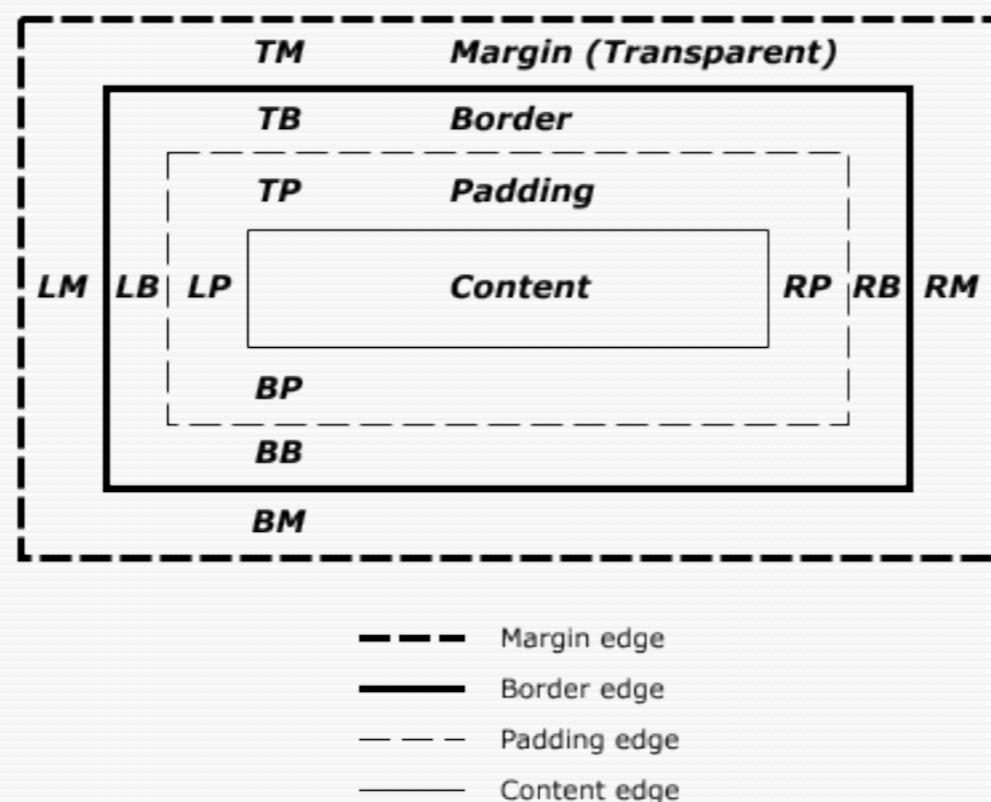
CALCULATION

WIDTH

width + padding-left + padding-right + border-left + border-right

HEIGHT

height + padding-top + padding-bottom + border-top + border-bottom



CyberAgent

The screenshot shows the homepage of the CyberAgent website. At the top, there's a navigation bar with links like 'Cyber', 'GridOverlay', 'RWD', 'Perf', 'Add to CodeBox', 'Baseliner', 'gist.io bookmarklet', '本番 [Jenkins]', 'STG [Jenkins]', and 'Other Bookmarks'. Below the header is the CyberAgent logo and a main menu with categories: ニュース (News), 技術情報 (Technology), 事業情報 (Service), 会社情報 (Corporate), 採用情報 (Recruit), and 投資家情報 (Investor Relations). A search bar is also present. The main content area features a large banner with the text 'ベンチャーキャピタル' (Venture Capital) and 'CyberAgent'.

Developer Tool

The screenshot shows the developer tools interface with several panels open. The top navigation bar includes tabs for Elements, Resources, Network, Sources, Timeline, Profiles, Audits, Console, and PageSpeed. The left sidebar displays a hierarchical DOM tree. The right side features a Metrics panel showing the dimensions and padding of a specific element. The Metrics panel has sections for Computed Style, Styles, and Metrics. The Metrics section shows a bounding box with a width of 468 and a height of 42, with a padding of 10 on all sides. Other panels visible include a Press Release section for a Candy app release and a Smartphone Service section.

FLOAT



CyberAgent

LAYOUT

Floatはいまだにcssにおけるレイアウトの基本プロパティ

(<http://jsbin.com/abixih/3/edit>)



CyberAgent

CLEARFIX

```
.clearfix:before,  
.clearfix:after {  
    content: ". . .";  
    display: block;  
    height: 0;  
    overflow: hidden;  
}  
.clearfix:after {  
    clear: both;  
}  
.clearfix {  
    zoom: 1; /* IE < 8 */  
}
```



CyberAgent

CLEARFIX

```
.cf:before,  
.cf:after {  
    content: " ";  
    display: table;  
}  
.cf:after {  
    clear: both;  
}  
.cf {  
    *zoom: 1;  
}
```



CyberAgent

LIST



CyberAgent

UNORDERED

```
<ul>
  <li>iPad</li>
  <li>iPhone</li>
  <li>MacBook Air</li>
</ul>
```

ORDERED

```
<ol>
  <li>iPad</li>
  <li>iPhone</li>
  <li>MacBook Air</li>
</ol>
```

DEFINITION

```
<dl>
  <dt>リスト見出し</dt>
  <dd>リスト意味</dd>
</dl>
```

NEST

```
<ol>
  <li>Walk the dog</li>
  <li>Fold laundry</li>
  <li>Go to the grocery and buy:
    <ul>
      <li>Milk</li>
      <li>Bread</li>
      <li>Cheese</li>
    </ul>
  </li>
  <li>Mow the lawn</li>
  <li>Make dinner</li>
</ol>
```

(<http://jsbin.com/ocipul/1/edit>)



CyberAgent

No.
151

ARTICLES • TOPICS • ABOUT • CONTACT • CONTRIBUTE • FEED

SEPTEMBER 27, 2002

CSS Design: Taming Lists

by MARK NEWHOUSE

Published in: CSS, HTML and XHTML, Layout

[Discuss this article »](#) [Share this article »](#)

As early as July of 1999 I was pontificating on email lists about the virtues of style sheets. Some things never change.

What *has* changed is how I think about CSS, and the underlying structure of (X)HTML to which it is applied. For example, I find that most pages on the web contain a menu of links in a navigation area. These are often marked up as a string of links, often in separate DIVs or paragraphs. Structurally, however, they are a list of links, and should be marked up as such.

Of course the reason that we don't mark them up in that way is that we don't want a bullet in front of every link in our navigation area. In a [previous article](#) I outlined several techniques for using CSS to layout a web page. One of those techniques involved manipulating a list to display horizontally rather than vertically.

In this article, I'll demonstrate how to use CSS to bring unwieldy lists under control. It's time for you to tell lists how to behave, instead of letting them run wild on your web page.

Search ALA

GO

 Include discussions

Topics

[Code >](#)[Content >](#)[Culture >](#)[Design >](#)[Mobile >](#)[Process >](#)[User Science >](#)

Snapshot

Do you crave the disciplined order of proper (X)HTML lists but long for control over their presentation? You can put a stop to their wild ways and bad behavior. Mark Newhouse shows you how to



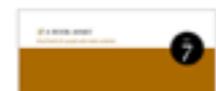
Finally! Things 2 brings

<http://www.alistapart.com/articles/taminglists/>

```
<li>Item 1</li>
<li>Item 2</li>
<li>Item 3</li>
<li>Item 4</li>
<li>Item 5 we'll make a bit longer so
    that it will wrap</li>
</ul>
```

More on the Job Board.

A Book Apart



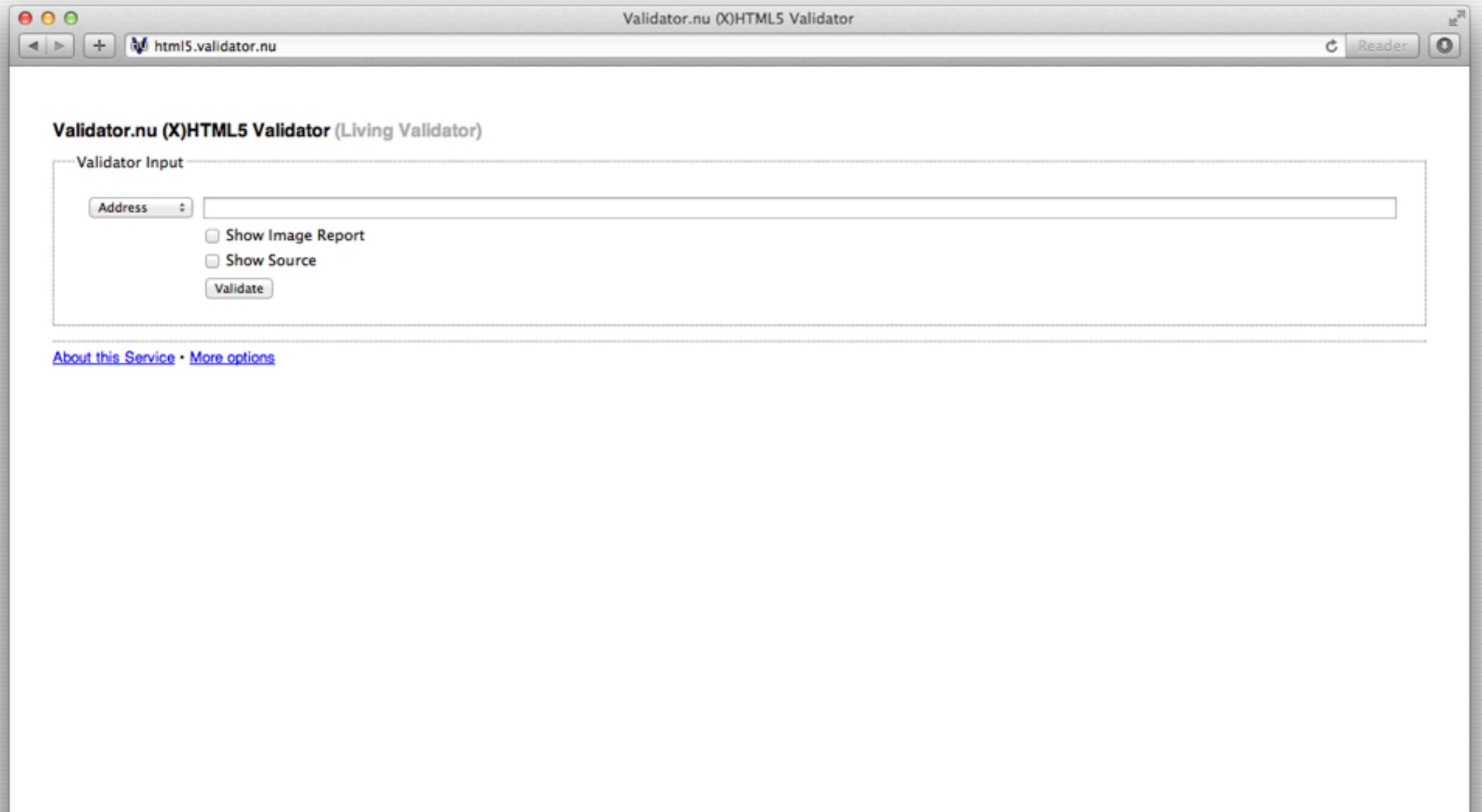
BEST PRACTICE



CyberAgent

HTML

スタンダードの遵守



<http://html5.validator.nu/>

スタンダードの遵守

```
<p id="intro">  
<strong>Lorem ipsum dolor sit amet.</p></strong>  
<p id="intro">Ut enim ad minim veniam, quis  
nostrud exercitation.
```

間違いに気がつきましたか？



CyberAgent

スタンダードの遵守

```
<p class="intro"><strong>Lorem ipsum dolor sit amet.</strong></p>
<p class="intro">Ut enim ad minim veniam, quis nostrud exercitation.</p>
```

そんなミスはしない。と思っている人も多いと思いますが
タグの閉じ忘れや閉じ位置の間違いですら
かなり多いです。



セマンティック要素の利用



CyberAgent

About HTML semantics and front-end architecture by Nicolas Gallagher | en.ja | Translated Articles for Front End Developers

enja.studiomohawk.com/2012/03/20/about-html-semantics-and-front-end-architecture/ Reader

en.ja
Translated Articles for Front End Developers

Home / About

About HTML semantics and front-end architecture by Nicolas Gallagher

20 Mar 2012

Credit

This article is translated with permission of Nicolas Gallagher(@necolas).
Original article is written at nicolasgallagher.com.
Titled: [About HTML semantics and front-end architecture – Nicolas Gallagher](http://nicolasgallagher.com/about-html-semantics-and-front-end-architecture/)

本記事はNicolas Gallagher(@necolas)氏の了承を得て翻訳された記事です。
原文はnicolasgallagher.comに書かれた、[About HTML semantics and front-end architecture – Nicolas Gallagher](http://nicolasgallagher.com/about-html-semantics-and-front-end-architecture/)にあります。

<http://tinyurl.com/sementics-ja>

私自身が気に入っている考え、体験、アイデアのコレクションでありかつ、1年間実験しつづけているアイデアのコレクション。
記事はHTMLのセマンティックについて、コンポーネントについて、フロントエンドの構造についてのアプローチ、クラスの命名パターン、そしてHTTPの圧縮について。

セマンティックとは

特にチームでの開発、運用フェーズを考慮した開発において、同意された意味が發揮する力は大きいです。

HTMLには100を超える要素があり、
それらを使い分けるのは思っている以上に大変です。



HTMLはウェブサイト、ウェブアプリケーションにおける基礎となる言語です。

この後につづく**CSS**、そして**JavaScript**は**HTML**に強い依存関係を持ちます。

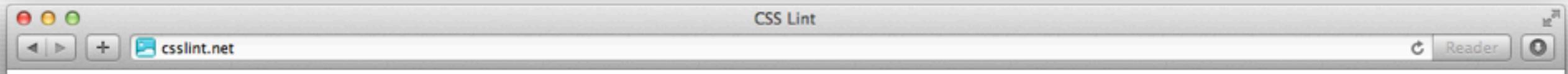
そういう観点で**HTML**を設計、実装するように心がけてください。

css



CyberAgent

スタンダードの遵守



CSS LINT

Will hurt your feelings*
(And help you code better)

Your CSS goes here. The more, the better. Linting works best when we see the big picture, so give us everything you've got.

LINT!

<http://csslint.net>

コメントの活用



CyberAgent

コメントの活用

```
/* =====  
セクションコメントブロック  
===== */  
  
/* サブセクションコメントブロック  
===== */  
  
/* コメント */
```



CyberAgent

コメントの活用

```
/*  
## ブロック名
```

説明文

...

HTML

例: <button class="btn primary">Primary</button>

...

*/



CyberAgent

docs | StyleDocco

jacobrask.github.com/styledocco/styledocco/examples/styledocco/docs.html

StyleDocco docs Search

StyleDocco default styles

This is the default stylesheet for StyleDocco. It also serves as an example of the documentation syntax.

Main layout

The page content has a maximum width of 1500 pixels.

When StyleDocco encounters a first level heading, the documentation is split into a new section with a pair of documentation and CSS code.

Documentation

CSS code

```
* {  
    -moz-box-sizing: border-box;  
    box-sizing: border-box;  
    margin: 0;  
    padding: 0;  
    border: 0;  
}  
body {  
    /* Top/bottom paddings for bars */  
    padding: 60px 0 40px;  
    background-color: hsl(207, 5%, 89%);  
    color: hsl(207, 5%, 30%);  
}  
.container {  
    max-width: 1300px;  
    margin: 0 auto;  
    padding: 0 20px;  
}  
.section {  
    position: relative;  
    margin-bottom: 20px;  
}  
.docs {  
    position: relative;  
    z-index: 2;  
    width: 68%;  
    min-height: 200px;  
    background-color: hsl(207, 0%, 100%);  
    background-clip: padding-box;  
    border: 1px solid hsla(207, 5%, 5%, .1);  
    border-radius: 5px;  
    box-shadow: 0 0 3px hsla(207, 5%, 5%, .1);  
}  
.code {  
    position: absolute;  
    top: 5px; bottom: 5px;  
    right: 0;  
    z-index: 1;  
    width: 33%;  
    padding: 10px 10px 10px 20px;  
    border-radius: 0 5px 5px 0;  
    border: 1px solid hsla(207, 5%, 5%, .1);  
    background-color: hsla(207, 5%, 95%, .75);  
    background-clip: padding-box;  
    opacity: .5;  
}
```

```
<article class="section">  
  <div class="docs">  
    <h1>Documentation</h1>  
  </div>
```

<http://jacobrask.github.com/styledocco/>

be automatically updated.

The width of the previews can be adjusted to test responsive designs. The height will be automatically adjusted to fit the content. Free hand resizing currently only works properly in Firefox. In WebKit browsers you can only enlarge the previews and other browsers have not implemented CSS `resize`.

Editable code blocks have a desaturated orange background. They will be automatically resized to fit the text inside.

```
.preview iframe {  
    display: block;  
    width: 100%;  
    height: 100%;  
    overflow: hidden;  
}  
.preview-code {  
    display: block;  
    width: 100%;  
    height: 100%;  
}
```

宣言ブロックの書式

```
.selector-1,  
.selector-2,  
.selector-3[type="text"] {  
    -webkit-box-sizing: border-box;  
    -moz-box-sizing: border-box;  
    box-sizing: border-box;  
    display: block;  
    font-family: helvetica, arial, sans-serif;  
    color: #333;  
    background: #fff;  
    background: linear-gradient(#fff, rgba(0, 0, 0, 0.8));  
}
```



CyberAgent

モジュラーCSS

About HTML semantics and front-end architecture by Nicolas Gallagher | en.ja | Translated Articles for Front End Developers
enja.studiomohawk.com/2012/03/20/about-html-semantics-and-front-end-architecture/ Reader

en.ja
Translated Articles for Front End Developers

Home / About

About HTML semantics and front-end architecture by Nicolas Gallagher

20 Mar 2012

Credit

This article is translated with permission of Nicolas Gallagher(@necolas).
Original article is written at nicolasgallagher.com.
Titled: [About HTML semantics and front-end architecture – Nicolas Gallagher](http://nicolasgallagher.com/about-html-semantics-and-front-end-architecture/)

本記事はNicolas Gallagher(@necolas)氏の了承を得て翻訳された記事です。
原文はnicolasgallagher.comに書かれた、[About HTML semantics and front-end architecture – Nicolas Gallagher](http://nicolasgallagher.com/about-html-semantics-and-front-end-architecture/)にあります。

<http://tinyurl.com/sementics-ja>

私自身が気に入っている考え、体験、アイデアのコレクションでありかつ、1年間実験しつづけているアイデアのコレクション。
記事はHTMLのセマンティックについて、コンポーネントについて、フロントエンドの構造についてのアプローチ、クラスの命名パターン、そしてHTTPの圧縮について。

効率的なセレクタ

// Bad Example

ul.news li a {...}

ul.news li a em.special {...}

// Good Example

.news a {...}

.news .special {...}



CyberAgent

効率的なセレクタ

ifなどの条件文が存在しない**CSS**と**HTML**にとって、
CSSのセレクタは条件文のように振る舞う。
例外処理が多くなればなるほどメンテナンスは困難に
なりやすいため、効率的にセレクタを設計する必要がある。

idiomatic-css/translations/ja-JP at master · necolas/idiomatic-css · GitHub

GitHub, Inc. github.com/necolas/idiomatic-css/tree/master/translations/ja-JP Reader

github

Signup and Pricing Explore GitHub Features Blog Sign in

PUBLIC [necolas / idiomatic-css](#)

Star 1,538 Fork 158

Code Network Pull Requests 2 Issues 3 Graphs

branch: master Files Commits Branches 1 Tags Downloads

[idiomatic-css / translations / ja-JP](#)

name	age	message	history
..			
README.md	2 months ago	Fix typo and rename dir 'sr' to 'sr-SR' [necolas]	

[translations/ja-JP/README.md](#)

一貫性のあるCSSらしいCSSを書くための原則

以下の文書はCSS開発のための合理的なスタイルガイドの骨子です。規範になるべきものではないし、私自身のスタイルの好みを他者のコードに押し付けるつもりは一切ありません。しかし、これらのガイドラインは現存する一般的で分別のあるパターンを利用することを推奨します。

このドキュメントは常に変わっていくものなので、新しいアイデアを常に求めています。ぜひアイデア等を投稿してください。

一貫性のあるCSSらしいCSSを書くための原則の英語版はこちらから

<http://tinyurl.com/idiomatic-css-ja>

- 6. 実例
- 7. 編成
- 8. ビルドとデプロイ

謝辞

FRONTEND PACKAGE



CyberAgent

SIMON COLLISON

PRESENTS

DEVELOPING YOUR

★★ **ULTIMATE** ★★

PACKAGE

<http://tinyurl.com/frontend-package>

HTML5 ★ BOILERPLATE

A rock-solid default for HTML5 awesome.

- I'm new here: plz explain why it's good, first.

[DOWNLOAD BOILERPLATE 3.0.2 UPDATED FEB 19TH](#)

[DOWNLOAD
BOILERPLATE](#) OR [DOWNLOAD
BOILERPLATE "STRIPPED"](#) OR [CUSTOMIZE
BOILERPLATE](#)

KEEP THE HINTS AND LINKS

NO COMMENTS, JUST THE BIZNASS.

100% HIPSTER.

[READ THE DOCS](#) ★ [CONTRIBUTE ON GITHUB](#) ★ [FOLLOW ON TWITTER](#)

HTML5 BOILERPLATE IS 3.0!

MONDAY, FEB 6TH, 2012

The key feature of this update is making Boilerplate smaller. Most of the changes have been working towards that. We have significant work done to our build script, so much so that we thought it deserved its own repository. Meanwhile, here is what's new.

CHANGELOG

3.0 gets some major updates [Here](#) are all the commits since last release.

[Should you upgrade existing sites?](#) Short answer: nah.. you're good.

<http://html5boilerplate.com>

* Respond.js is no longer available by default.

[INDEX.HTML](#)

* Chromeframe no longer applied via script. IE6 users are served links to [install Chromeframe](#) or visit

JAVASCRIPT



CyberAgent

基本構文

```
var name = Joe;  
var age = 45;  
var job = plumber;  
  
alert( name );
```



CyberAgent

QUESTION

キーワードVarがある場合と
そうでない場合の違いを簡潔に説明してください。



CyberAgent

タイプ(型)

- ▶ String
- ▶ Numbers
- ▶ Booleans
- ▶ Null
- ▶ Undefined
- ▶ Object
- ▶ Array
- ▶ Data

演算式

```
var four = 2 + 2,  
    // 加算  
hour = 24 - 13,  
    // 減算  
seventy = 7 * 10,  
    // 乗算  
avg_days_per_week = 365 / 52,  
    // 除算  
remainder = 31 % 2,  
    // 余り  
msg = "the time is " + hour + " o'clock";  
    // stringsを加算;
```

<http://jsbin.com/ububop/1/edit>



```
var num = 10;  
num += 5; // same as num = num + 5  
alert(num); // 15  
num -= 3; // same as num = num - 3  
alert(num); // 12  
num *= 2; // same as num = num * 2  
alert(num); // 24  
num /= 6; // same as num = num / 6  
alert(num); // 4  
num %= 3; // same as num = num % 3  
alert(num); // 1
```



- ▶ **false**
- ▶ **null**
- ▶ **undefined**
- ▶ **"" (空文字)**
- ▶ **0**
- ▶ **Nan**

FALSE



CyberAgent

```
var a = "a string";
alert( !a ); // false
alert( !!a ); // true
```



CyberAgent

条件式

```
var year = 2011, msg = "In the ";
if (year > 2000) {
    msg += "twenty-first century";
} else {
    msg += "twentieth century";
}
console.log(msg); // In the twenty-first century
```



CyberAgent

QUESTION

```
var order_size = "medium";
switch (order_size) {
  case "small":
    alert("small");
  case "medium":
    alert("medium");
  case "large":
    alert("large");
  case "extra large":
    alert("extra large");
  default:
    alert("something else?");
}
```

<http://jsbin.com/eyipiz/1/edit>

ループ

```
var names = ["Ringo", "John", "Paul", "George"];
for (var i = 0; i < names.length; i++) {
  alert("Say hello to " + names[i]);
}
```



CyberAgent

<http://jsbin.com/itogim/1/edit>



CyberAgent

```
// for-in loop
var person = {
    name : "Jules Verne",
    job  : "Author",
    year_of_birth: 1828,
    year_of_death: 1905
},
prop;

for (prop in person) {
    alert("His " + prop + " is " + person[prop]);
}
```



スタイル / 形式



CyberAgent

idiomatic.js/translations/ja_JP at master · rwldrn/idiomatic.js · GitHub

GitHub, Inc. github.com/rwldrn/idiomatic.js/tree/master/translations/ja_JP Reader

github

PUBLIC  [rwldrn / idiomatic.js](#)

Signup and Pricing Explore GitHub Features Blog Sign in

Code Network Pull Requests 5 Issues 9 Graphs

branch: master Files Commits Branches 5 Tags Downloads

[idiomatic.js / translations / ja_JP](#)

name	age	message	history
..			
 readme.md	2 months ago	Adds Sonar JavaScript plugin to code quality tools. Closes #66 [Godin]	

 [translations/ja_JP/readme.md](#)

一貫性のある慣用的なJavaScriptの書き方

これは継続しているドキュメントであるため、私たちのコードを改善できる新しいアイデアを常に歓迎します。貢献方法: **fork**、**clone**、**branch**、**commit**、**push**、**pull request**。

• Rick Waldron [@rwaldron](#), [github](#)

<http://tinyurl.com/idiomatic-js-ja>

- Breno Oliveira [@garu_rj](#), [github](#)
- Leo Beto Souza [@leobetosouza](#), [github](#)
- Ryuichi Okumura [@okuryu](#), [github](#)
- Pascal Precht [@PascalPrecht](#), [github](#)
- EngForDev [engfordev](#) - Hwan Min Hong / MinTaek Kwon [@leoinsight](#) / Tw Shim [@marocchino](#), [github](#) / Nassol Kim [@nassol99](#), [github](#)

関数、タイプメソッド



CyberAgent

閾数



CyberAgent

```
function square(number) {  
    return number * number;  
}
```



CyberAgent

```
var square = function(number) {  
    return number * number  
};
```



CyberAgent

関数 or メソッド？



CyberAgent

```
objectName.methodname = function_name;
```

```
var myObj = {  
    myMethod: function(params) {  
        // ...do something  
    }  
};
```



CyberAgent

引数

```
function getGreeting(hour) {  
    hour = hour || new Date().getHours();  
    if (hour < 12) {  
        return "Good Morning!";  
    } else if (hour < 17) {  
        return "Good Afternoon!";  
    } else {  
        return "Good Evening!";  
    }  
}  
alert( getGreeting( new Date().getHours() ) );
```



```
var msg1 = getGreeting(10),  
    msg2 = getGreeting(1,2,3,4,5,6, "and on"),  
    msg3 = getGreeting(),  
    msg4 = getGreeting("five o'clock");
```



CyberAgent

ARGUMENTS OBJECT



CyberAgent

```
function myConcat(separator) {  
  var result = "", // initialize list  
  i;  
  // iterate through arguments  
  for (i = 1; i < arguments.length; i++) {  
    result += arguments[i] + separator;  
  }  
  return result;  
}  
  
myConcat("", "", "red", "orange", "blue");  
myConcat(; "", "elephant", "giraffe", "lion", "cheetah");  
myConcat(. "", "sage", "basil", "oregano", "pepper",  
"parsley");
```

<http://jsbin.com/utekap/1/edit>

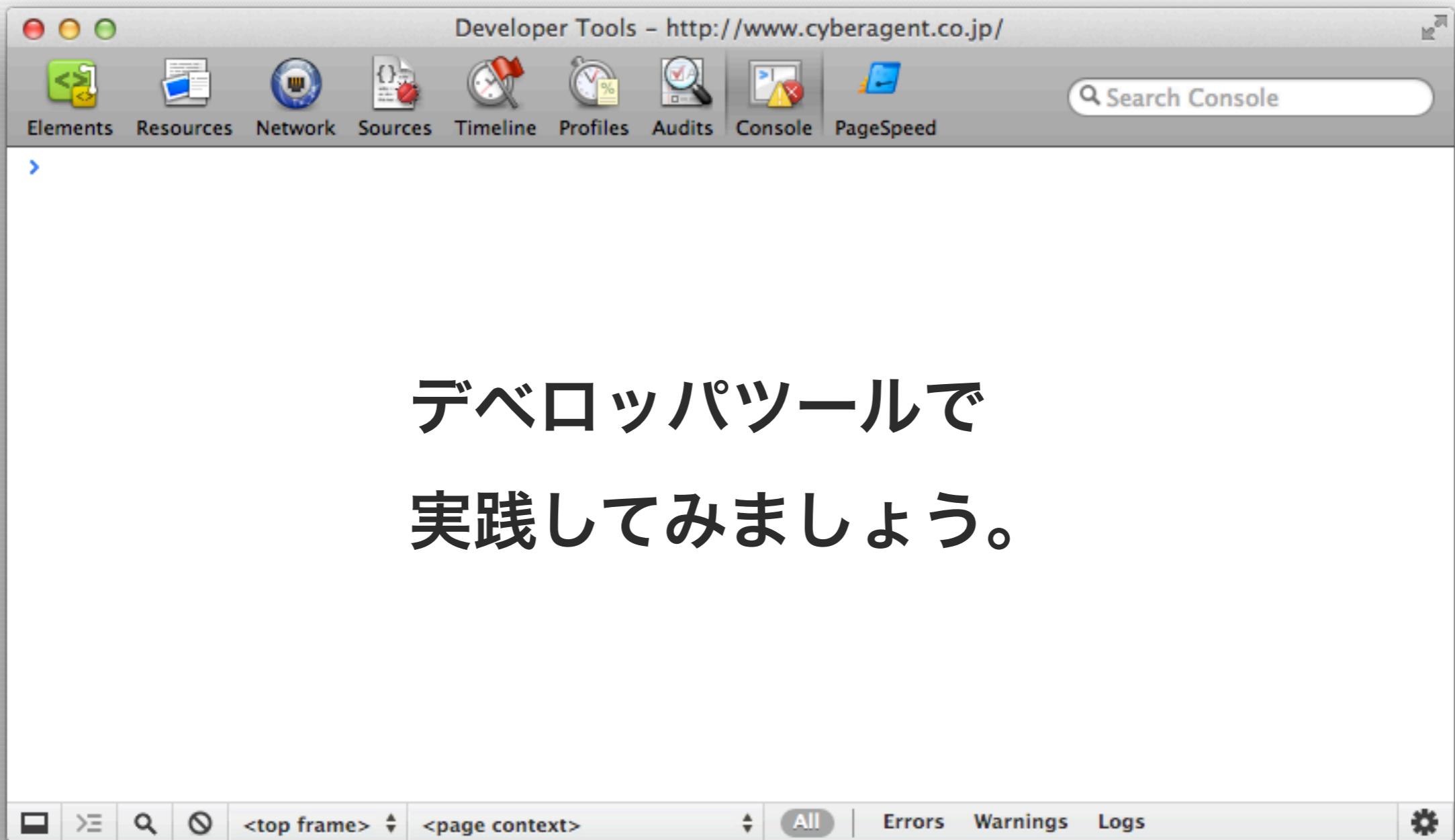


スコープ



CyberAgent

FUNCTION SCOPE



CyberAgent

BLOCK SCOPE

<http://jsbin.com/uxecew/1/edit>



CyberAgent

HOISTING

<http://jsbin.com/irahus/1/edit>



CyberAgent

無名関数

```
function () {  
    return "this function has no name";  
}  
;
```

```
var hi = function () {  
    return "Hi";  
};
```

```
var hi = (function () {  
    return function () {  
        return "Hi";  
    }  
}());
```



```
(function(){ /* code */ })();
```



CyberAgent

BEN ALMAN

[Home](#) | [News](#) | [Music](#) | [Photo](#) | [Portfolio](#) | [Projects](#) | [About](#)[All](#) | [Code](#) | [Game](#) | [Geek](#) | [Music](#) | [Personal](#) | [Photo](#) | [Project](#) | [Work](#)

Immediately-Invoked Function Expression (IIFE)

By "Cowboy" Ben Alman on November 15, 2010 4:33 PM | [Comments](#)

In case you hadn't noticed, I'm a bit of a stickler for terminology. So, after hearing the popular, yet misleading, JavaScript term "self-executing anonymous function" (or self-invoked anonymous function) one too many times, I've finally decided to organize my thoughts into an article.

In addition to providing some very thorough information about how this pattern actually works, I've actually made a recommendation on what we should call it, moving forward. Also, if you want to skip ahead, you can just check out some actual [Immediately-Invoked Function Expressions](#), but I recommend reading the entire article.

Please understand that this article isn't intended to be an "I'm right, you're wrong" kind of thing. I'm genuinely interested in helping people understand potentially complex concepts, and feel that using consistent and accurate terminology is one of the easiest things that people can do to facilitate understanding.

So, what's this all about, anyways?

In JavaScript, every [function](#), when invoked, creates a new execution context. Because variables and functions defined within a function may only be accessed inside, but not outside, that context, invoking a function provides a very easy way to create privacy.

```
// Because this function returns another function that has access to the
// "private" var i, the returned function is, effectively, "privileged."
function makeCounter() {
    // 'i' is only accessible inside 'makeCounter'.
    var i = 0;

    return function() {
        console.log( ++i );
    }
}
```

```
counter2(); // logs: 2
i; // ReferenceError: i is not defined (it only exists inside makeCounter)
```

In many cases, you won't need multiple "instances" of whatever your [makeWhatever](#) function returns, and can make do with just a single instance, and in other cases, you're not even explicitly returning a value.

 Search

PLEASE DONATE!

Your [generous donation](#) encourages me to continue developing and updating my code!

FEATURED PROJECTS

jQuery: [BBQ](#), [Bookmarklet Generator](#), [cond](#), [doTimeout](#), [equalizeBottoms](#), [getObject](#), [hashchange event](#), [iff](#), [longUrl](#), [Message Queuing](#), [Misc plugins](#), [outside events](#), [postMessage](#), [replaceText](#), [resize event](#), [Star Wipe](#), [throttle / debounce](#), [Utils](#), [unwrap](#), [URL Utils](#), [urlinternal](#)

JavaScript: [Battlelog Hacks](#), [Bookmarklets](#), [Debug](#), [Emotify](#), [Library Boilerplate](#), [Linkify](#)

Other: [Finder Copy Open Window Paths](#), [Multi-Firefox Launcher](#), [Safari View Source in TextMate](#), [Simple PHP Proxy](#), [Simplified](#)

RECENT NEWS

[Why grunt? Why not something else?](#)
[Fixing "Retina" Image Scaling with Gyazo](#)
[Multiple var statements in JavaScript](#)
[News Flash: Semicolons Required In](#)
[Introducing Grunt](#)

STAY UPDATED

[Subscribe to my news \(email\)](#)
[Subscribe to my photos \(email\)](#)

<http://tinyurl.com/benalman-iife>

タイプメソッド



CyberAgent

Activate Your Web Pages

第6版



JavaScript

リファレンス

O'REILLY®
オンライン・ジャパン

David Flanagan 著
木下 哲也 訳

JavaScriptリファレンス
by David Flanagan



CyberAgent

- ▶ **String**メソッド
- ▶ **Numbers**メソッド
- ▶ **Data**メソッド
- ▶ **Array**メソッド
- ▶ **Math**関数

Stringメソッド

- ▶ **length** (正確にはプロパティ)
- ▶ **indexOf**
- ▶ **slice / substr / substring**
- ▶ **split**

Numbersメソッド

- ▶ **toFixed**
- ▶ **parseInt** (正確にはNumberメソッドではない)
- ▶ **parseFloat** (正確にはNumberメソッドではない)



CyberAgent

Data×ソッド

- ▶ **get__**
- ▶ **set__**



CyberAgent

Arrayメソッド



CyberAgent

- ▶ **join**
- ▶ **pop / shift**
- ▶ **push / unshift**
- ▶ **reverse**
- ▶ **slice**
- ▶ **sort**

Math関数

- ▶ **min**
- ▶ **max**
- ▶ **random**
- ▶ **round / ceil / floor**

DOM



CyberAgent

WHAT IS DOM

HTML文書の構造、内容をプログラムから取り扱いやすくするための仕様が**DOM(Document Object Model)**です。

DOMとはHTMLドキュメントやXMLドキュメントをプログラムから利用するAPIで、それらをオブジェクトのツリー状の集合として取り扱います。

このツリーを**DOMツリー**と呼びます。



CyberAgent

WHAT IS NODE

DOMツリーの中のオブジェクトがノードと呼ばれます。
ノードはツリー構造になっているため、
ある1つのノードから他のノードを参照するときは
親ノード、子ノード、兄弟ノード、先祖ノード、子孫ノードと
呼ぶ事になります。

DOM Scripting:
Web Design with JavaScript and
the Document Object Model



DOM Scripting 標準ガイドブック

やさしく学ぶ、JavaScriptとDOMによるWebデザイン

Jeremy Keith (著) 吉川典秀 (訳) 中村享介 (株式会社ロクナカ) (監修)

DOM Scripting 標準ガイドブック

by Jeremy Keith

<http://www.amazon.co.jp/dp/4839922373>



CyberAgent

JQUERY



CyberAgent

"I had the moment when I realized **jQuery was essentially a "find something, do something" library.** I already knew CSS, and jQuery used CSS selectors for the "find something" part. "Do something" in jQuery can be as simple as "click", "hide", "show", "slideToggle", etc."

- Chris Coyier

FINDING ELEMENT



CyberAgent

- ▶ **getElementById**
- ▶ **getElementsByClassName (IE9以降)**
- ▶ **getElementsByTagName**
- ▶ **querySelectorAll (IE7以降)**
- ▶ **querySelector (IE7以降)**



CyberAgent

PERFORMANCE

<http://jsperf.com/getelementbyid-vs-queryselector/17>



CyberAgent

MODIFYING ELEMENT



CyberAgent

- ▶ **document.createElement();**
- ▶ **document.createTextNode();**
- ▶ **appendChild();**
- ▶ **insertBefore();**
- ▶ **removeChild();**
- ▶ **replaceChild();**
- ▶ **setAttribute();**
- ▶ **getAttribute();**
- ▶ **element.style**



CyberAgent

イベント

WHAT IS EVENT



CyberAgent

EVENTLISTENER

```
element.addEventListener  
(イベント, ハンドラ, useCapture);
```

EVENTLISTENER

```
var btn = document.querySelector('a');

btn.addEventListener('click', function (e) {
  console.log('btn: ' + this);
}, false);
```



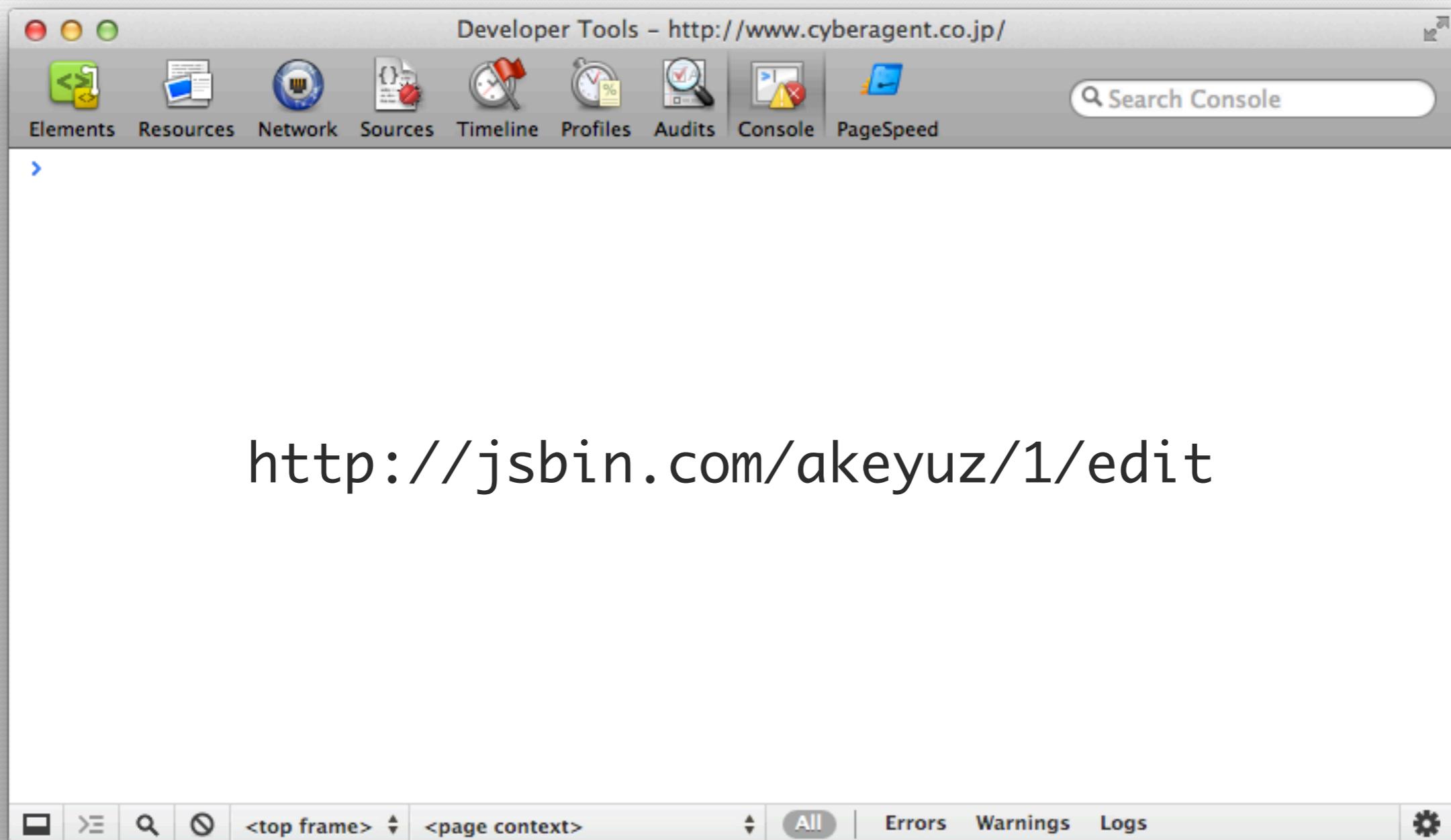
CyberAgent

EVENT PROPERTIES



CyberAgent

PREVENT



CyberAgent

TARGET



CyberAgent

Event delegation

- [MDN](#)
- [HTML5 Doctor](#)
- [HTML5 Rocks](#)
- [Expressive Web](#)
- [CreativeJS](#)

Source:

```
var resources = document.querySelector('#resources'),  
    log = document.querySelector('#log');
```

<http://tinyurl.com/eventdeleglation>

```
}  
if (target.tagName === 'LI') {  
  log.innerHTML = 'A list item';  
}  
if (target.tagName === 'UL') {
```

イベントの伝播

```
<html>
  <body>
    <div id="foo">
      <button id="bar">サンプル</button>
    </div>
  </body>
</html>
```



CyberAgent

キャプチャリングフェーズ

Windowオブジェクトからはじまり、
DOMツリーを下にたどってイベントが
伝播していくフェーズ



CyberAgent

ターゲットフェーズ

イベントターゲットに登録されている
イベントリストが実行されるフェーズ。



CyberAgent

バブリングフェーズ

イベントターゲットから**DOMツリー**を
上にたどって行くフェーズ。

最終的には**Window**オブジェクトまで
イベントは伝播される。



CyberAgent

CALCULATOR



CyberAgent

午前中に作成したフロントエンドパッケージを
コピーして、`/calculator`に名前を変更して、
計算機を作ってみましょう。

<http://cl.ly/IzbX>

課題

githubグループの**/assignment/**内に
assignment-week1.mdファイルがあります。
グループのリポジトリを**fork**して、
/assingment/内に自分の名前のディレクトリ**/FIRST_LAST/**
を作成し、回答したファイルを**Pull Request**してください。

WEEK 1 END



CyberAgent