



jQuery高速化

to-R 西畑 一馬

自己紹介

西畑 一馬

to-R代表 Webクリエイター



KazumaNishihata

- ・ JavaScript / Ajax システム開発
- ・ HTML5 / CSS3 スマートフォンサイト作成



西畑一馬のjQuery Mobileデザイン入門

ASCII.jp : 西畑一馬のjQuery Mobileデザイン入門

http://ascii.jp/elem/000/000/607/607366/

ASCII.jp × WEB PROFESSIONAL

Google カスタム検索

RSSフィード メールマガジンサービス

ASCII.jp 特別レビュー

“夢の壁掛け”を実現できる?

40型で10kg以下??
テレビの新しいスタイルを提案!

「フリースタイルAQUOS F5」の実力に迫る!!

トップ ビジネス TECH WEB PROFESSIONAL デジタル Mac/iPod ゲーム・ホビー 自作PC

価格比較 Microsoft キャリア クラウド アキバ スマートフォン オーディオ&ビジュアル PanaSpot 富士通 DELL Lenovo EPSON SONY McAfee 住まい情報局

はてブ注目記事

- XHTML1.0とHTML5の違いをスタバのページで紹介 547 users
- サンプルで学ぶjQuery Mobileまとめ (後編) 31 users
- CSS3 AnimationsとJSの連携でキャラクターを制御 24 users
- グーグル、中小企業が無料でウェブサイトを作成できる「みんなのビジネスオンライン」 提供開始 9 users
- iPhone/iPad両対応! PhoneGapで作るアルバムアプリ 8 users
- 40分で覚える! jQuery速習講座 2048 users
- 絶対見ておきたい有名企業のスマホサイトまとめ 2024 users
- もう配色には困らない! 「Adobe Kuler」 1649 users
- 30分でできる! Webサイトを高速

[PR] 光対応マンションへ引越し!! 快適なPCライフを実現! 住まい情報局

西畑一馬のjQuery Mobileデザイン入門

いまもっとも注目されているモバイルフレームワーク「jQuery Mobile」。jQuery Mobileの基本的な使い方から、実践的なデザインテクニックまで。iPhone/Androidなどのスマートフォンサイト制作にjQuery Mobileを活用する方法を具体的に解説します。

文●西畑一馬/to-R

2011年08月19日 13時00分

第7回 サンプルで学ぶjQuery Mobileまとめ (後編)

モバイルフレームワーク「jQuery Mobile」のグリッドレイアウト、開閉パネル、フォーム関連機能の基本的な使い方をサンプルコードとともに紹介する。

2011年08月04日 11時00分

第6回 サンプルで学ぶjQuery Mobileまとめ (中編)

モバイルフレームワーク「jQuery Mobile」のリスト表示、ヘッダー/フッターなどの基本的な機能を簡単なサンプルとともにまとめて紹介する。

2011年07月28日 13時00分

第5回 サンプルで学ぶjQuery Mobileまとめ (前編)

注目のモバイルフレームワーク「jQuery Mobile」のスマートフォンサイ

jQuery Mobile

ヘッダー

選択してください

1回

2回

3回

ページ1

One

Three

Five

Seven

コンテンツ1

ヘッダー

キーワードで絞り込み

関連記事

- jQuery作者が明かす「jQuery Mobile」の魅力
- ジョン・レッシングに聞く、jQueryのこれから
- Dreamweaver CS5.5でスマホアプリ作ってみた
- 実践! iPhone&Androidサイト制作ガイド
- 40分で覚える! jQuery速習講座
- スマホ対応の新潮流「レスポンシブWebデザイン」とは?
- NimbleKitでiPhone用動画アプリを自作しよう
- Canvasで作ったWebアプリをiPhoneアプリに変換
- HTML5でiPhoneアプリ開発! PhoneGap入門
- PhoneGapでiPhoneのカメラアプリを作ろう

本日開講

http://ascii.jp/elem/000/000/607/607366/

著書



本日の内容

- 高速化の基本
- ベンチマーク
- セレクタの高速化
- DOM操作の高速化
- Ajaxの最適化
- DOM Readyイベント
- まとめ

高速化の基本

高速化の基本

ブロック	A	B	C	D
コスト	54%	4%	30%	12%

続・ハイパフォーマンスWebサイトより

遅い箇所にポイントを絞って高速化

遅くなりがちなポイント

- ループが高コスト
- DOM操作が高コスト

高速化の例



http://www.asahibeer.co.jp/superdry/extracold_app/

高速化の例

- XMLから店舗情報を取得してGoogle Mapに配置
- 店舗は250件前後

```
$.each($(".shoplist",data),function(){
    shopList.push({
        "shopname":$(".shopname",this).text(),
        "lat":$(".nl",this).text(),
        "lng":$(".el",this).text(),
        "detailurl":$(".detailurl",this).text(),
        "position": new google.maps.LatLng($(".nl",this).text()-0,
            $(".el",this).text()-0)
    });
});
```

高速化の例

- XMLから店舗情報を取得してGoogle Mapに配置
- 店舗は250件前後

```
$.each($(".shoplist",data),function(){  
    shopList.push({  
        "shopname":  
        "lat":$(".lat",this).text(),  
        "lng":$(".el",this).text(),  
        "detailurl":$(".detailurl",this).text(),  
        "position": new google.maps.LatLng($(".nl",this).text()-0,  
            $(".el",this).text()-0)  
    });  
});
```

Android2.1で15秒くらい

高速化の例

- eachメソッドからfor文に変更

```
var shoplistArr = $("shoplist",data).get(),key;
for (key in shoplistArr){
    shop = shoplistArr[key];
    shopList.push({
        "shopname":$("shopname",shop).text(),
        "lat":$("nl",shop).text(),
        "lng":$("el",shop).text(),
        "detailurl":$("detailurl",shop).text(),
        "position": new google.maps.LatLng($("nl",shop).text()-0,
            $("el",shop).text()-0)
    });
}
```

高速化の例

- eachメソッドからfor文に変更

```
var shoplistArr = $("shoplist",data).get(),key;  
for (key in shoplistArr){  
    shop = shoplistArr[key];  
    shopList.push(  
        "shopname":shopname,shop).text(),  
        "lat":$("nl",shop).text(),  
        "lng":$("el",shop).text(),  
        "detailurl":$("detailurl",shop).text(),  
        "position": new google.maps.LatLng($("nl",shop).text()-0,  
            $("el",shop).text()-0)  
    });  
}
```

Android2.1で6秒くらい

高速化の例

- DOMアクセスから正規表現に変更

```
data = data.replace(/\n/g, "");
var regexp = new RegExp(/<shopname><!\[CDATA\[ (.+?) \]><\/shopname>.+?<nl>( .+?)<\/nl>.+?<el>( .+?)<\/el>.+?<detailurl><!\[CDATA\[ (.+?) \]><\/detailurl>/g),m;
while (m = regexp.exec(data)) {
    shopList.push({
        "shopname":m[1],
        "lat":m[2]-0,
        "lng":m[3]-0,
        "detailurl":m[4],
        "position": new google.maps.LatLng(m[2]-0,m[3]-0)
    });
}
```


高速化の例

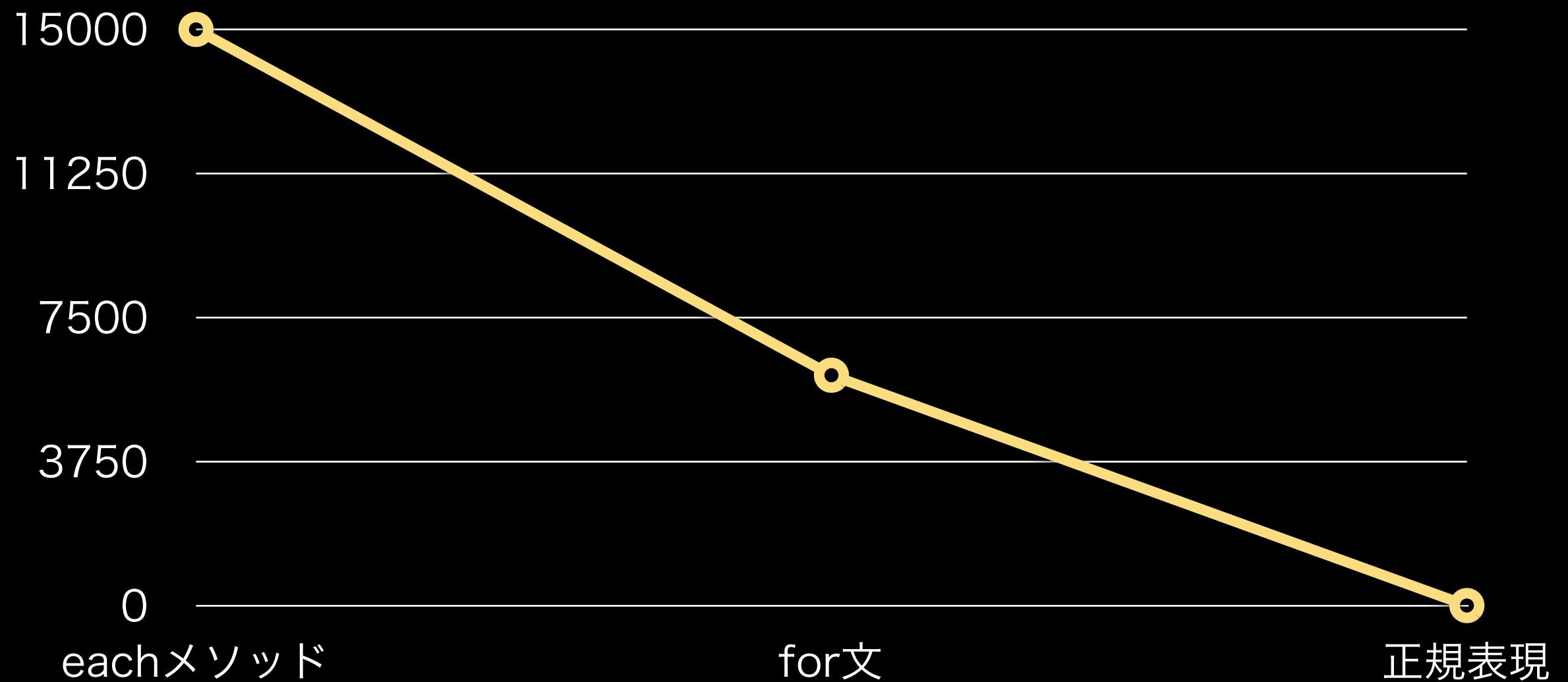
- DOMアクセスから正規表現に変更

```
data = data.replace(/\n/g, "");
var regexp = new RegExp(/<shopname><!\[CDATA\[([.+]?)\]\]><\/
shopname>.+?<nl>([.+]?)<\/nl>.+?<el>([.+]?)<\/el>.+?<detailurl><!\[
CDATA\[([.+]?)\]\]><\/detailurl>.+?<\/>/);
while (m = regexp.exec(data)) {
    shopList.push({
        "shopname":m[1],
        "lat":m[2]-0,
        "lng":m[3]-0,
        "detailurl":m[4],
        "position": new google.maps.LatLng(m[2]-0,m[3]-0)
    });
}
```

Android2.1で6ミリ秒くらい

高速化の例

○ 処理時間



ベンチマーク

測定方法

- Firebug

```
$(function(){  
    console.time("id");  
    $("#tbl td").each(function(){  
        $(this).css("background","red");  
    });  
    console.timeEnd("id");  
    //id:4msと出力される  
});
```

Google Chrome / Safariでも利用可能

測定方法

- getTime()を利用した測定方法

```
$(function(){  
    start=(new Date()).getTime();  
    $("#tbl td").each(function(){  
        $(this).css("background","red");  
    });  
    alert((new Date()).getTime()-start);  
    //4と出力される  
});
```

セレクトタの高速化

セレクタの高速化

```
$("#table td").css("background" , "red");
```

(中略)

```
$("#table td").css("border" , "1px solid red");
```

(中略)

```
$("#table td").css("margin" , "100px");
```

セレクタのキャッシュ

```
var tbl = $("table td");  
tbl.css("background" , "red");  
(中略)  
tbl.css("border" , "1px solid red");  
(中略)  
tbl.css("margin" , "100px");
```

メソッドチェーン

```
$("#table td").css("background" , "red")  
  .css("border" , "1px solid red")  
  .css("margin" , "100px");
```

コンテキストの利用

```
$("#table td a").css("background" , "red");  
$("#table td span").css("background" , "red");
```



```
var tbl = $("#table td")  
$("#a",tbl).css("background" , "red");  
$("#span",tbl).css("background" , "red");
```

遅くなることもあるので注意！！

ID属性の利用

```
$("#tbl td a").css("background" , "red");  
$("#attention1").css("background" , "red");
```

重要

querySelectorAll

querySelectorAll

- Internet Explorer 8+
- Firefox 3.5+
- Opera 10+
- Chrome 1+
- Safari 3.2+
- iPhone
- Android

jQueryのセレクトタ判別

querySelectorAll
が使える

Yes



querySelectorAll

No



Sizzleエンジン

querySelectorAll

X `$("#query .cssnite:first")`

○ `$("#query .cssnite:first-child")`

△ `$("#query .cssnite").filter(":first")`

CSSセレクタ以外利用しない

IE8での注意点

- CSSセレクタの対応状況が悪い

<http://nekodama.sakura.ne.jp/css/sSupport.php>

X `$("#query .cssnite:last-child")`

- querySelectorAllのつもりが通常のSizzleエンジンになっ
てしまうことも・・・
- querySelectorAllのパフォーマンスが悪い

レガシーブラウザの最適化



Internet Explorer 6



Internet Explorer 7

Sizzleエンジン

- セレクタをスペースやコロンで分割
- 順番に絞り込み検索を行う

```
$("#query .cssnite:first")
```



```
$("#query").find(".cssnite").filter(":first")
```

レガシーブラウザの最適化

- findメソッドの活用
- class属性には要素を追加
- シンプルなセレクトタにする
- Traversingメソッドの活用

X `$("#query .cssnite:first")`

○ `$("#query").find("div.cssnite").first()`

Traversingメソッド

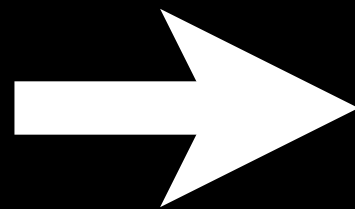
```
$("#div:first")
```

```
$("#div:last")
```

```
$("#a:not(.ex)")
```

```
$("#a:has(img)")
```

```
$("#tr:eq(5)")
```



```
$("#div").first()
```

```
$("#div").last()
```

```
$("#a").not(".ex")
```

```
$("#a").has("img")
```

```
$("#tr").eq(5)
```

結局どうすればいいの？

- ターゲットブラウザを明確に！
- 遅いブラウザに対して最適化する
- ベンチマークを測定

DOM操作の高速化

DOM操作の高速化

```
var ul = $("ul");  
for(var i=0 ; i<50 ; i++){  
    ul.append("<li>" + i + "</li>");  
}
```

DOM操作の高速化

```
var ul = $("ul");  
var lis="";  
for(var i=0 ; i<50 ; i++){  
    lis += "<li>" + i + "</li>";  
}  
ul.append(lis);
```

DOMのアクセスは最小限に

DOM操作の高速化

```
var ul = $("ul");  
var lis = new Array;  
for(var i=0 ; i<50 ; i++){  
    lis.push("<li>" + i + "</li>");  
}  
ul.append(lis.join(""));
```

IE6では+=演算子で巨大な内容を扱うと遅くなる

DOM操作の高速化



<http://zaq.ne.jp/asobu/>

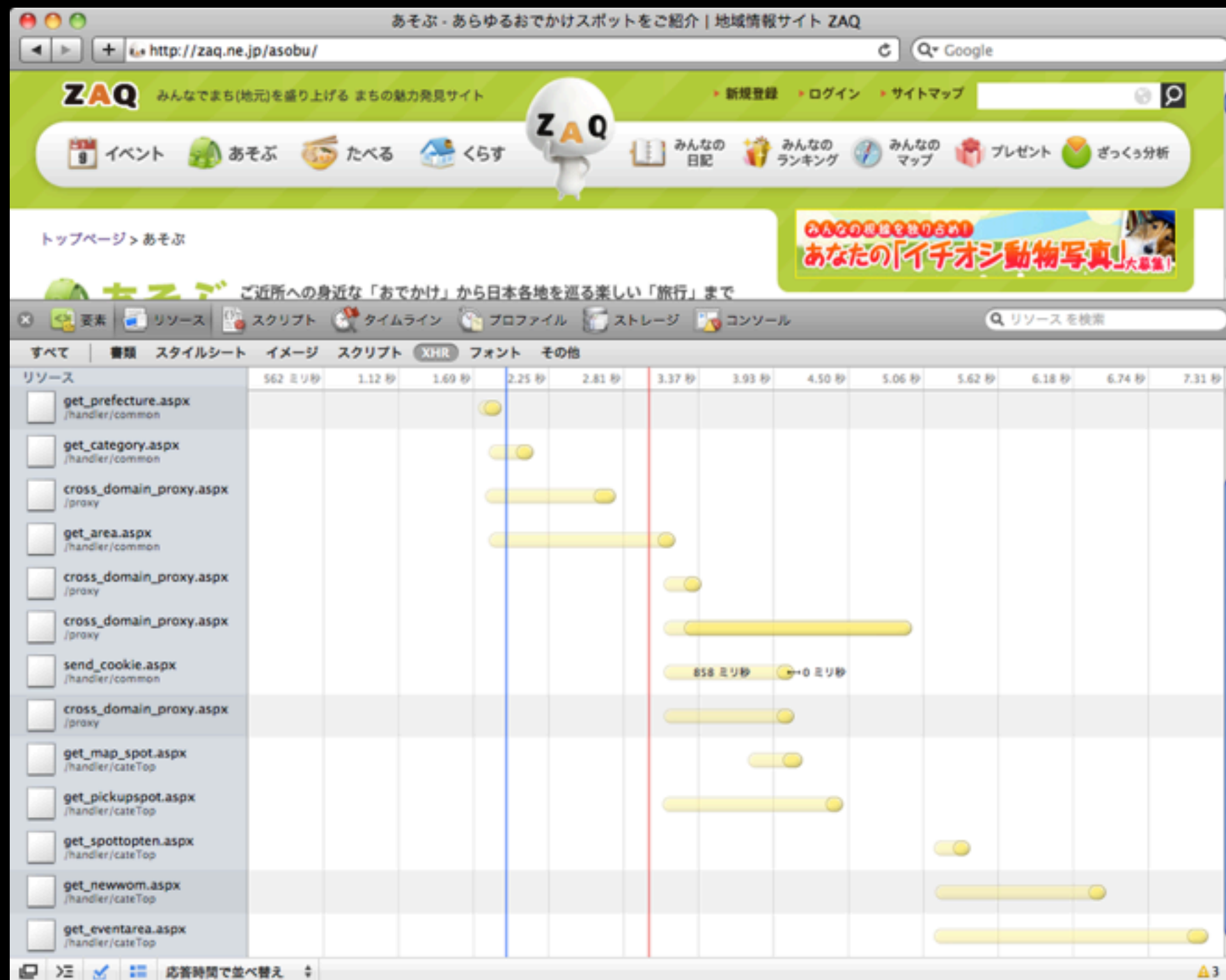
プロジェクトの副産物 - デスマなう -



<http://5509.me/log/deathmanow/>

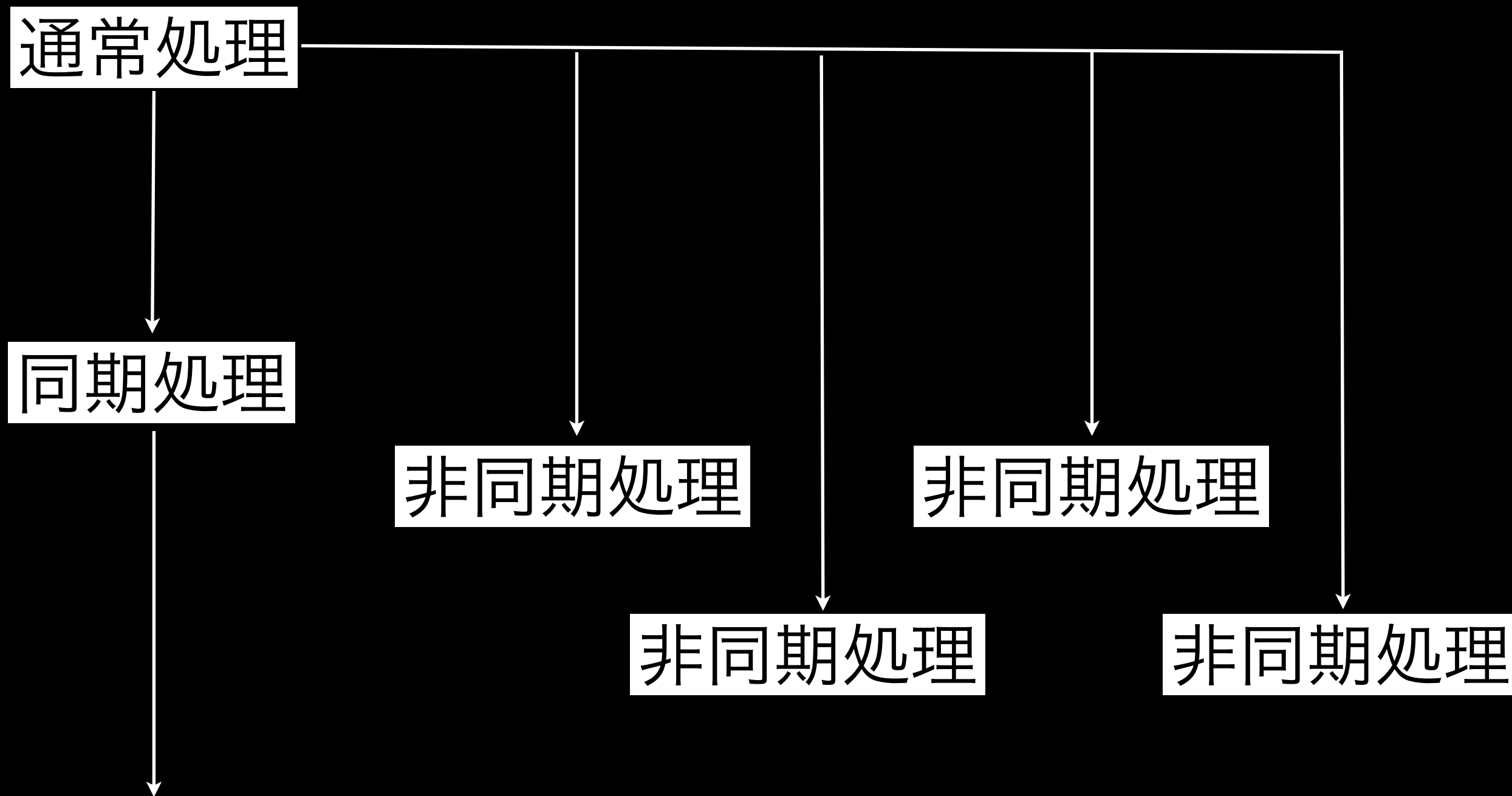
Ajaxの最適化

非同期通信のコントロール



表示するだけで14本のAjaxリクエスト！

非同期通信



Ajaxリクエスト

```
GEvent.addListener(map, 'moveend', function(func) {  
    $.ajax({  
        url : "xxx.json" ,  
        success : function(json){  
            //マップ処理  
        }  
    });  
});
```

Ajaxリクエストの数を減らす

```
var mapVer = 0;
GEvent.addListener(map, 'moveend', function(func) {
    mapVer++;
    var mapVerStart = mapVer;
    setTimeout(function(){
        if(mapVerStart == mapVer){
            $.ajax({....});
        }
    }, 500);
});
```

abortにより古いAjaxリクエストの破棄

```
var mapVer = 0 ,mapload=[];  
GEvent.addListener(map, 'moveend',function(func) {  
    if(mapload[mapVer])mapload[mapVer].abort();  
    mapVer++;  
    var mapVerStart = mapVer;  
    setTimeout(function(){  
        if(mapVerStart == mapVer){  
            mapload[mapVer] = $.ajax({....});  
        }  
    },500);  
});
```

success時も常に最新のAjaxをチェック

```
mapload[mapVer] = $.ajax({  
    url : requestUri ,  
    type : "POST" ,  
    success : function(json){  
        if(mapVerStart != mapVer) return false;  
        //マップ処理  
    }  
});
```

Ajaxのポイント

- サーバーサイドにかなりの負荷がかかる！
- こまめに不要な処理を削除
- 場合によってはサーバーサイドに
処理を振り分ける

DOM Readyイベント

DOM Readyイベント

```
$(function(){  
  //jQueryの命令を記述していく  
});
```

すべての要素が読み込み終わった後に実行される

body要素

```
<script>  
$("table td").css("background" , "red");  
</script>  
</body>
```

この段階でだいたいの要素の読み込みは終了している
AjaxなどDOM操作に関係ない処理は早い段階で行う

スコープの生成

```
<script>
(function($){
  var tbl;
  tbl = $("table td").css("background" , "red");
})(jQuery);
</script>
</body>
```

画像の読み込み後に実行

```
$(window).load(function(){  
    var height = $("div#css").height();  
});
```

```
var imgSize = $("div#css img").size();
var loadImages = function(){
    --imgSize;
    if(imgSize==0){
        var height = $("div#css").height();
    }
}

$("div#css img").load(function(){
    loadImages();
}).error(function(){
    loadImages();
});
```

確実に画像の読み込み後に実行

よいプログラムとは？

- 開発速度が速い？
- 再利用性が高い？
- メンテナンス性が高い？
- レスポンスが速い？

よいプログラムとは？

- 開発速度が速い？
- 再利用性が高い？
- メンテナ
- レスポンスが速い？

どれも重要!!!

バランスが一番大事

まとめ

- 高速化の基本は遅い処理の最適化
- 遅い処理はベンチマークをとって判別
- DOM操作やループ処理は遅くなりがち
- 遅いデバイス・ブラウザを最適化
- AjaxではAjaxリクエストを意識する
- イベントは必要最小限だけ設定する

ちょっとだけ告知

ロクナナワークショップ

西畑一馬のCSS3でつくるスマートフォンサイト講座

9月29日 / 10月21日 / 11月18日

<http://67.org/ws/workshop/detail/091.html>

END

Thank You!!