

# Firefox 3.1 in Depth<sup>(?)</sup>

@Firefox Developer Conference 2008

by Tomoya Asai (aka. dynamis)



# 自己紹介 - 浅井 智也

- Mozilla Japan テクニカルマーケティング担当
- Mozilla の製品、技術、思想のエバンジェリスト
- コミュニティでは dynamis (でゅなみす) として活動
- Firefox 1~3, Thunderbird 1~2 の L10N をリード
  - 単独リードからあさんや mar さんの共同体制に
- SUMO (Firefox サポートサイト) 日本語版リーダー
  - 実際の翻訳の大半は mar さんによるもの
- MozillaZine.jp などのコミュニティ活性化サポート
  - 基本的にはサーバなどのバックエンドサポート



今日の話題は…



# Shiretoko



知床





Photo by hogeasd

<http://www.flickr.com/photos/9177053@N05/2811388694/>

国立公園





Photo by mtsn

<http://www.flickr.com/photos/mtsn/242646868/>

世界遺產





Photo by s.sawada

[http://www.flickr.com/photos/s\\_sawada/340629324/](http://www.flickr.com/photos/s_sawada/340629324/)

キタキツネらぶ~



Amazonプライムの無料体験をご利用いただけます

Amazon プライム無料体験で、この商品について通常配送を無料でご利用いただけます。



[その他のイメージを見る](#)



[自分のイメージを掲載する](#)

## 知床ぬいぐるみ キタキツネ

[\[詳細\]](#)

まだカスタマーレビューはありません。今すぐどうぞ。[この商品の詳細](#)

参考価格: ~~¥ 1,050 (税込)~~

価格: **¥ 940 (税込)** 1500円以上国内送料無料  
外あり)でお届けします。[詳細](#)

OFF: **¥ 110 (10%)**

通常6~7日以内に発送します。在庫状況について詳しくは[こちら](#)  
この商品は、Amazon.co.jp が販売、発送します。ギフト包装を利用

Screenshot on Amazon.co.jp

<http://www.amazon.co.jp/dp/B000SOH366/>

可愛くない(>\_<)



それはさておき



# Firefox 3.1



# 本日のお料理

前菜: New Interface

スープ: New Feature

魚料理: New Standard

肉料理: New Technology

デザート: New Web





Photo by hiroohi

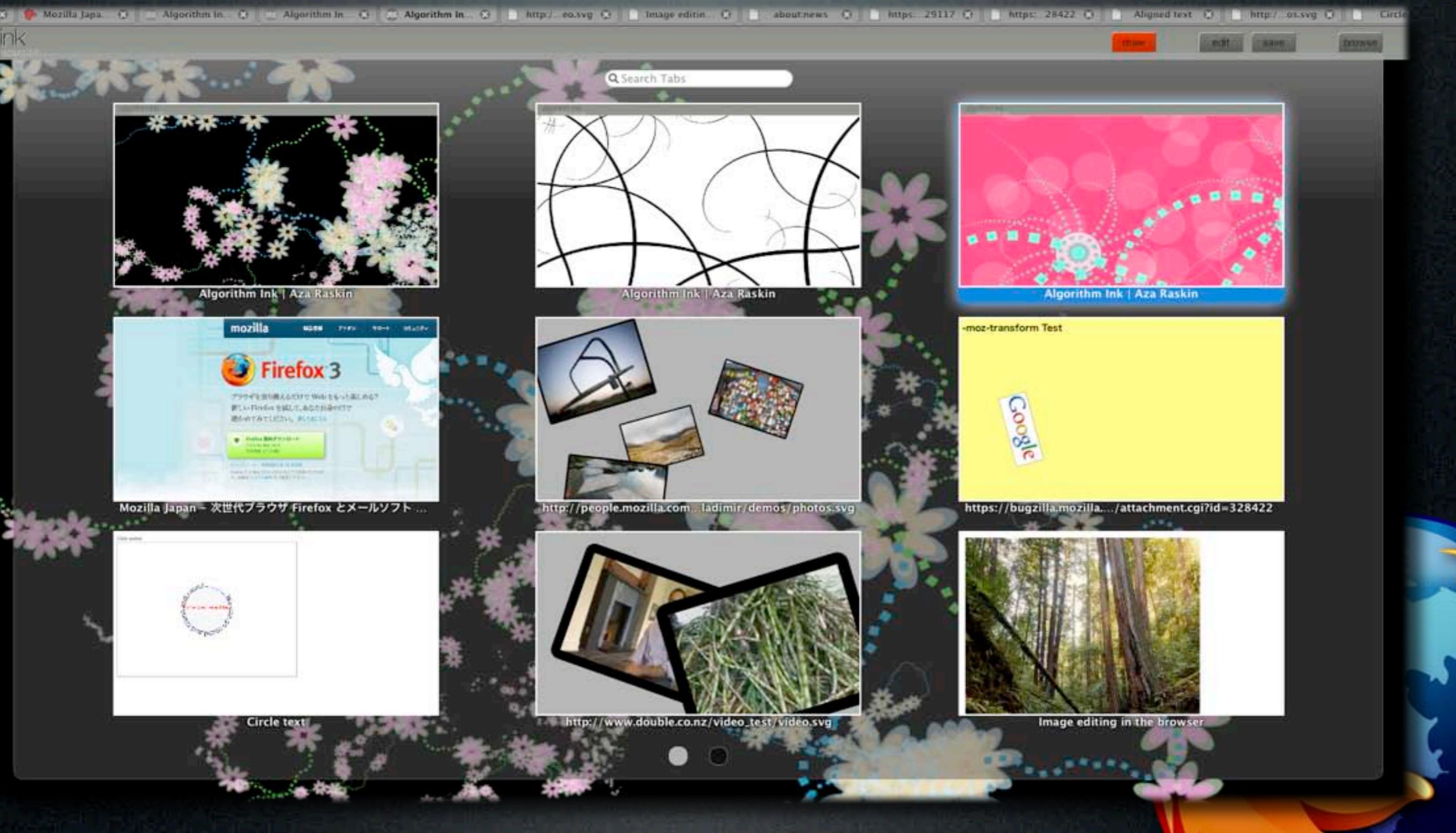
<http://flickr.com/photos/rainboweyes/2747484161/>

# New Interface



# タブのプレビュー

- Ctrl + Tab、タブ一覧ボタンでプレビュー表示と検索



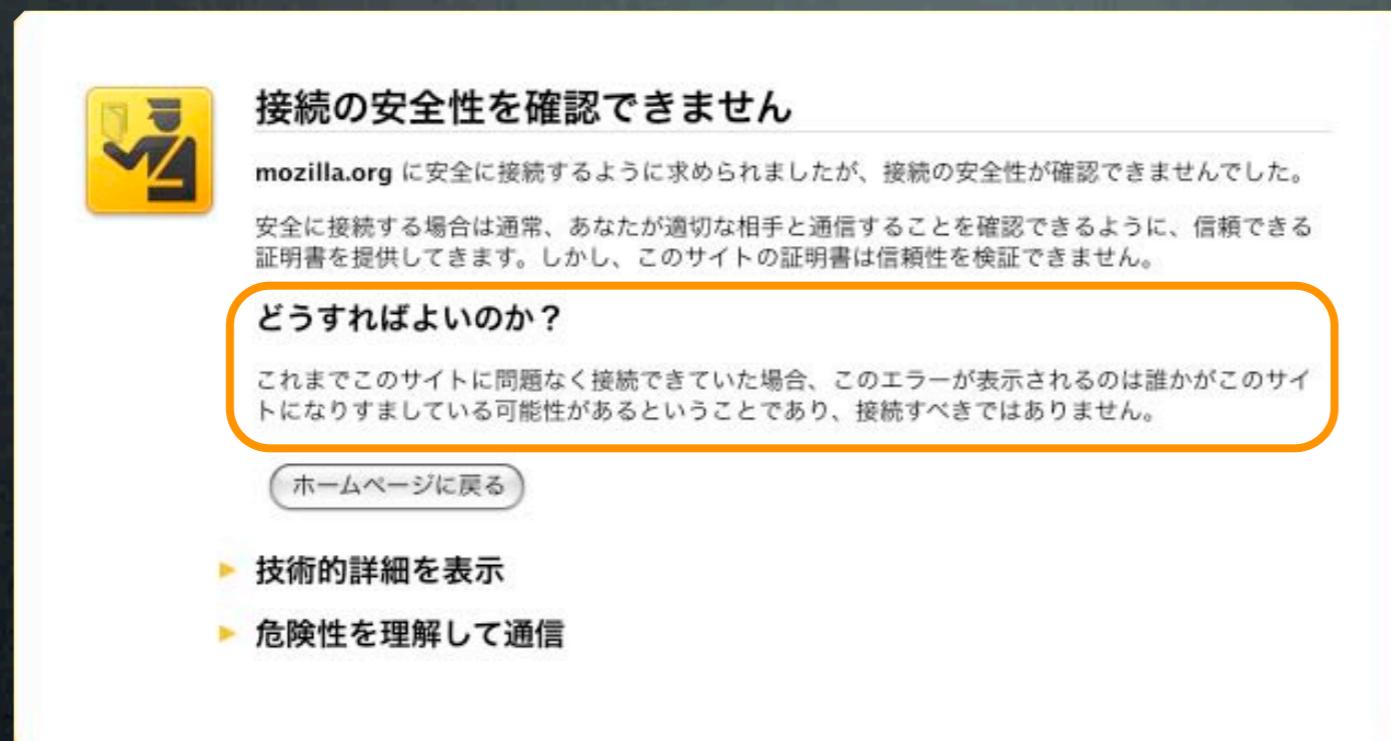
# タブブラウジング

- ドラッグ&ドロップでウィンドウ間のタブ移動
  - Fx3まではページが再読み込みされていた
- タブが1つでもデフォルトでタブバーを表示
  - タブ1つだけの時はタブに閉じるボタンを表示しない
  - 最後のタブを閉じるとウィンドウが閉じる
- タブバーに新しいタブを開くボタンを追加
- 再読み込みボタンの中クリックでタブを複製

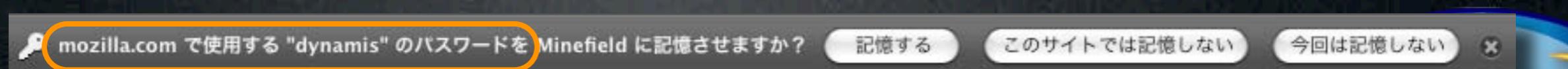


# セキュリティ関連 UI

- SSL 証明書エラーページを平易な説明に改善



- パスワード保存時の情報バーにドメイン名を表示



- ブロックされたプラグインについて情報バーで通知

# その他の UI 変更

- インストーラの EULA を廃止して about:rights に変更
  - オープンソースソフトで EULA は不要
- ロケーションバーが空の時には説明を表示
  - ブックマークと履歴を検索: 
- 読み込み中アイコン(スロバー)をデフォルト非表示に
- MacBook のマルチタッチジェスチャーをサポート
  - 履歴、ページ端への移動、ズーム、タブ切り替え





Photo by akira

<http://flickr.com/photos/arika/428018383/>

# New Feature



# プライベートブラウジング

- 表示ページの履歴、検索履歴、ダウンロード履歴、Web フォーム履歴、Cookie、一時ファイルが一切保存されない（匿名ブラウジングではない）
- モード切替時にタブの状態を保存、復元する
- ロケーションバーは既存の履歴とブックマークが対象

 **プライベートブラウジング**

Minefield はこのセッション中にはブラウズ履歴を一切保存しません。

Minefield はこのセッション中にはブラウズ履歴を一切保存しません。表示ページの履歴、検索履歴、ダウンロード履歴、Web フォーム履歴、Cookie、一時ファイルについては一切保存されませんが、ダウンロードしたファイルと作成したブックマークについては保存されます。

最近の履歴を消去して起動できます。

[最近の表示履歴を消去](#)

プライベートブラウジングを終了するにはツールメニューの“プライベートブラウジング”的チェックを外してください。



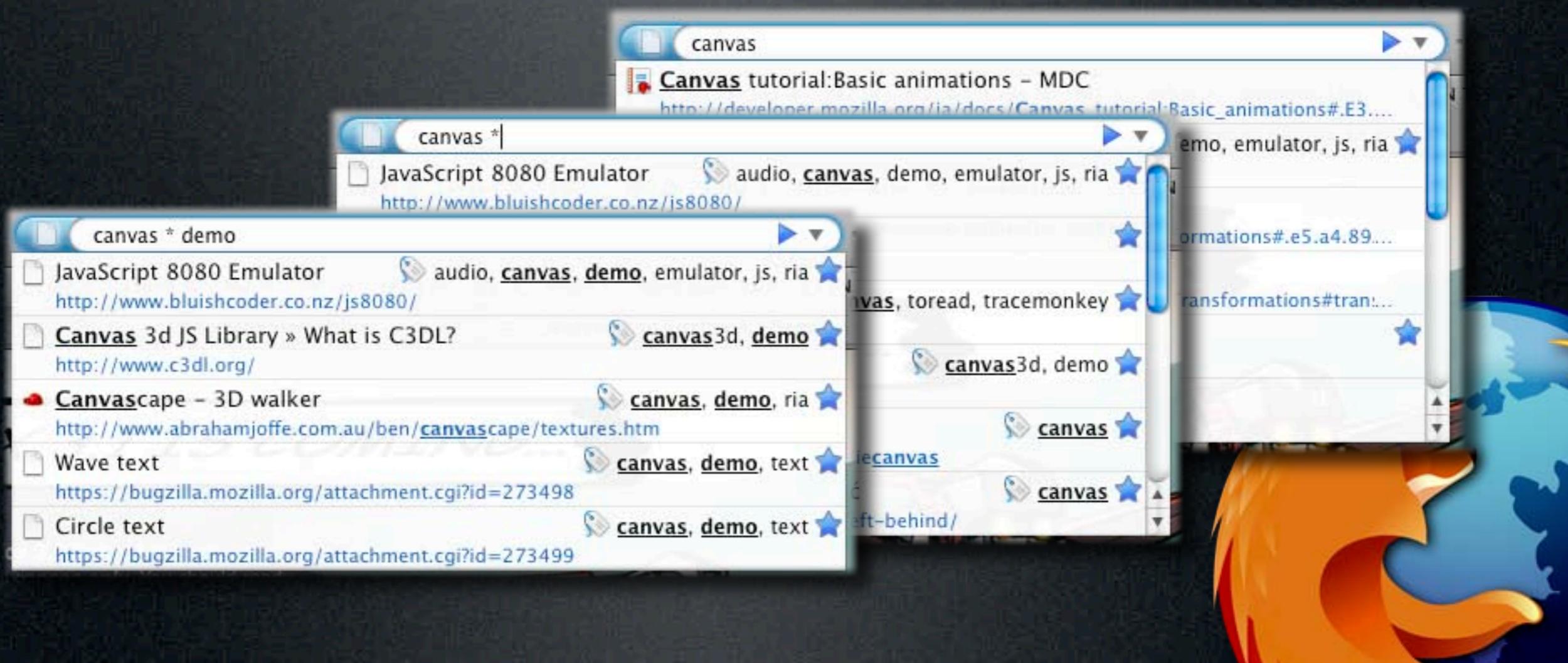
# 最近の履歴を消去

- 期間を指定して履歴を消去
  - 1, 2, 4時間、1日、すべての履歴から選択



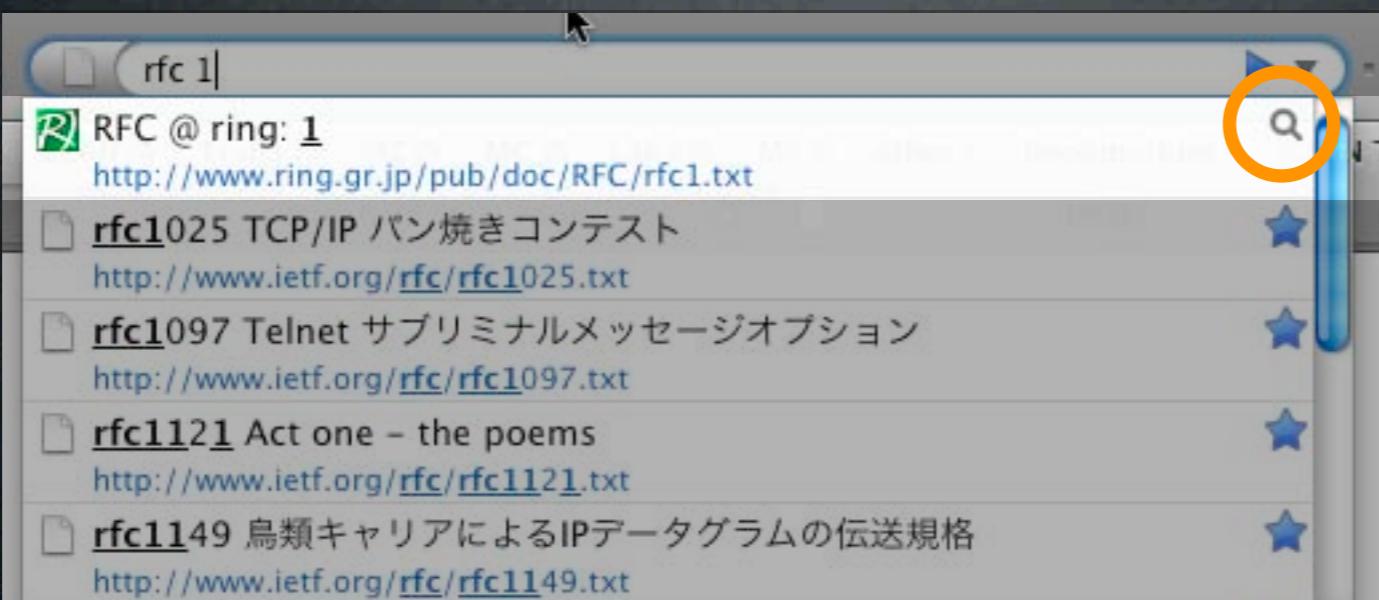
# スマートロケーションバー

- 検索対象の絞り込みキーワード（記号）のサポート
  - ^で履歴、\*でブックマーク、+でタグ付きページ  
@でURL、#でタイトルとタグだけを検索対象に
  - 注: "^word" ではなく "^ word" や "word ^" と入力



# スマートロケーションバー

- 検索文字列の途中を変更した時にも即時結果を表示
  - Fx3 では文字列末尾を変更したときだけ表示
- スマートキーワードの検索を候補のトップに表示
- キーワード "rfc" を設定したブックマークの例:



# 柔軟なセッション復元

- ・ ウィンドウとタブを指定して復元が可能に

前回終了時の状態を復元しますか？

Minefield は前回正常に終了されませんでした。ご迷惑をおかけして申し訳ありません。前回終了時のウィンドウとタブを復元するか、必要なければ状態を復元せず新しいセッションで起動してください。

Minefield が繰り返し異常終了してしまう場合:

- アドオンマネージャで最近追加した拡張機能を無効化してみてください。
- 問題の原因と思われる Web ページ以外の状態を復元してみてください:

復元 ウィンドウとタブ

- ✓ □ ウィンドウ 1
  - Welcome to Firefox 3.1 Beta 1
  - タイムライン - Mozilla Japanese L10N - Trac
  - Mozilla Japan - イベント情報 - Firefox Developers Conference 2008
- ✓ □ ウィンドウ 2
  - Algorithm Ink | Aza Raskin

状態を復元せずに起動 前回終了時の状態を復元



- ・ クラッシュの原因となるタブだけ外して復元を



# その他の新機能

- タグの自動補完
- 複数ブックマークの一括タグ付け
- ソース表示のリンクブラウズモード
  - view-source でリンクを辿りながらソース表示
- DNS プリフェッチ





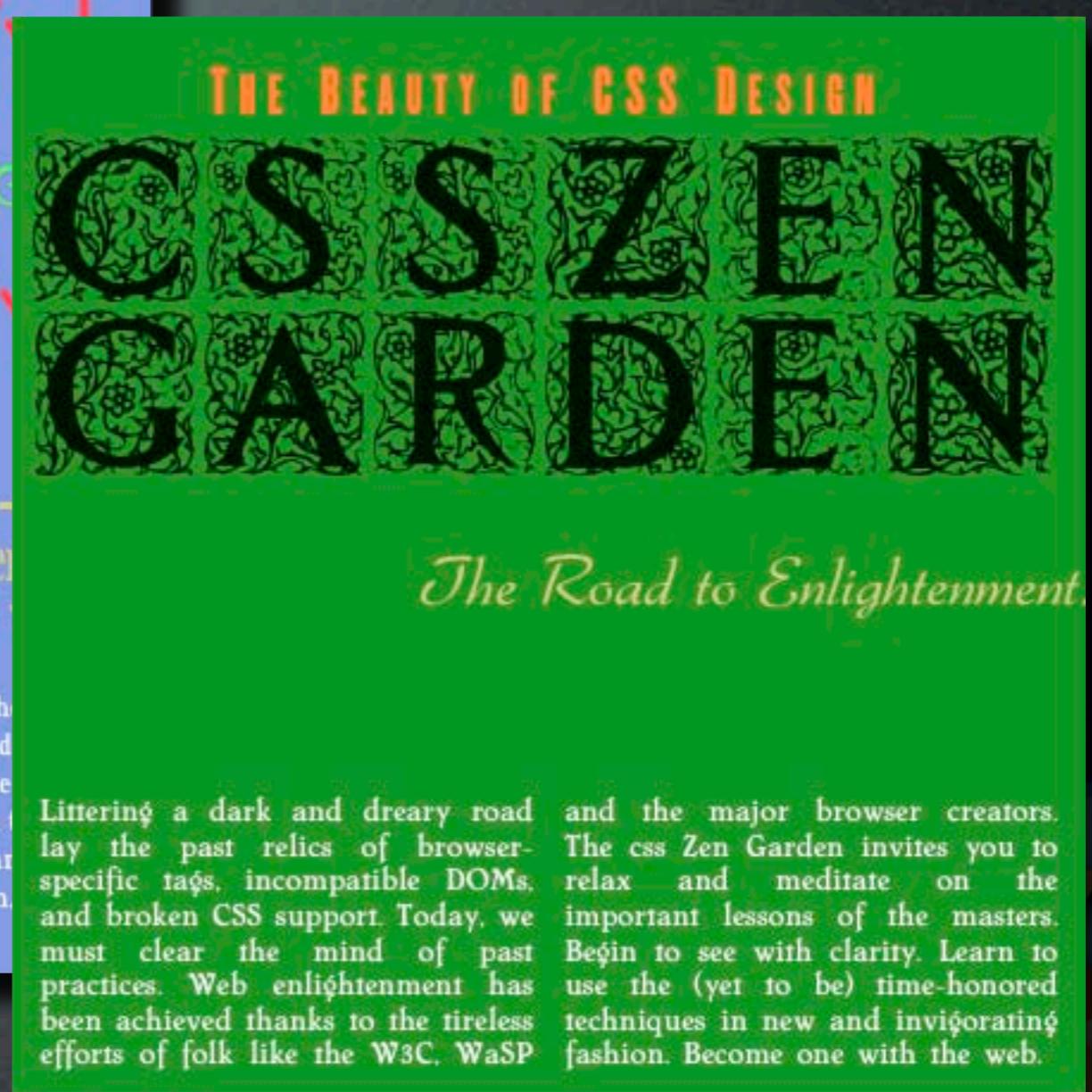
Photo by hiroohi

<http://flickr.com/photos/rainboweyes/2747484331/>

# New Standard



# Web Fonts



# Web Fonts

- ダウンロードフォントの font-face を定義して使う

```
@font-face {  
    font-family: "Decorated Roman Initials";  
    src: url(http://www.princexml.com/fonts/steffmann/DecoratedRomanInitials.ttf) format("truetype");  
}  
  
h1 {  
    font: 2.3cm/0.84 "Decorated Roman Initials", fantasy;  
}
```

- 必要な文字だけダウンロードさせることも可能

```
@font-face {  
    font-family: Ampersand;  
    src: url(ampersand.otf) format("opentype");  
    unicode-range:U+0026;  
}
```



# Web Fonts

- サポートしていないなければフォールバックするだけ
- フォント読み込み中の描画はブラウザによって異なる
  - Firefox 3.1 はダウンロード後に再描画
  - Safari はダウンロードまでテキスト表示無し
- フリーの Web Font も公開されている
- <http://opentype.info/demo/webfontdemo.html>
- フォントメーカーの皆さん、サイトライセンス如何？



# CSS3

- CSS3 メディアクエリー
  - 出力デバイスに応じたデザインが可能に
    - 画面の幅、高さ、縦横比、解像度、色深度 etc...
  - 特にネットブックやモバイルデバイス対応に
- CSS3 セレクタ(擬似クラス)のサポート:
  - 任意または特定種類の  $a_n+b$  番目の子要素
    - :nth-child()、:nth-last-child()、  
:nth-of-type()、:nth-last-of-type()
  - 最初、最後、单一の子要素
    - :first-of-type、:last-of-type、:only-of-type



# CSS3

- 楕円境界線 border-radius
- 影付き文字 text-shadow
- 影付きボックス (draft)  
-moz-box-shadow
- 画像境界線 (draft)  
-moz-border-image
- 段組レイアウト (draft) -moz-column-\*  
• オーバーフロー時に"..." text-overflow: ellipsis



モンは深き眠りに落ちた。蘇え  
りし野獸は群れを成し、地を覆  
い尽くさんばかりに広がった。人々は時  
の到来を贊美し、狡猾なる狐と共に実り  
を炎に捧げた。そして彼らは神聖なる預  
言の約束の地たる新しき世界を築き上  
げ、その子らに野獸のことを語り聞かせ  
た。眠りから目覚めしマモンよ、見よ！

もはや追従者でしかない己の姿を。

from The Book of Mozilla, 11:9



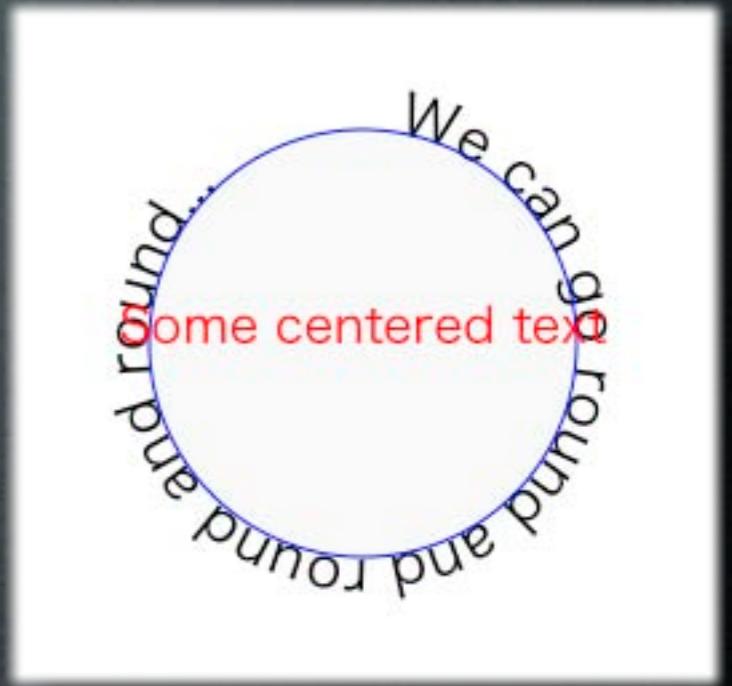
# more CSS

- 改行以外の空白を1つに CSS2.1 white-space:pre-line
- CSS2.1 generated content の position と float
- 変形、回転  
(Apple Spec) CSS Transform
- WebFonts、Transform、  
text-shadow のサポートで  
自由なテキストデザインが  
ブラウザ上で実現可能に
  - 斜めのメニュー や タイトル etc...



# Canvas

- HTML5 Canvas Text API
  - `fillText()`、`strokeText()`、`measureText()`,
  - Firefox 3 では独自 API
- HTML5 Canvas Shadows API
  - `ctx.shadowOffsetX`、`ctx.shadowOffsetY`、`ctx.shadowBlur`、`ctx.shadowColor`
- HTML5 Canvas CreateImageData()
- Canvas へのビデオ描画



# <video> & <audio>

- Ogg Vorbis/Theora をネイティブサポート
- SVG filter、clip、CSS Transform などとも併用可



# その他の Web 標準

- Geolocation API
- SVG による HTML の切り抜き、マスク、フィルタ
- HTML5 ドラッグ&ドロップ API
- HTML5 オフラインキャッシング
- DOM 2 Traversal - NodeIterator
- クロスサイト XMLHttpRequest (XHR)
  - Fx3 では仕様が未確定だったためスキップされた
- <script defer="defer"> で JavaScript の遅延評価





Photo by hiroohi

<http://flickr.com/photos/rainboweyes/2747484483/>

# New Technology



# カラーマネージメント

- ICC カラープロファイルの CMS がデフォルトでオンに
- PhotoShop などでの表示色と一致する
  - カラープロファイル情報のある画像でのみ色補正
  - 画像と背景色 (CSS 指定のRGB値) がズれないように
  - プロファイル付画像だけなのは W3C の指針でもある
  - これは先行実装している Safari とも同じ動作
- Fx3 では性能(と描画)の問題でデフォルトオフ
  - 速度低下は 1% (全画像だと 4~5%、Fx3 では 20~30%)

<http://bholley.wordpress.com/2008/09/12/so-many-colors/>



Firefox 3.0

Firefox 3.1



Photo by flob

<http://flickr.com/photos/flob/2850464994/>

# JavaScript

- コンテンツでのネイティブ JSON (rfc4627) サポート
  - JSON.parse()、JSON.stringify()、  
\*.prototype.toJSON()
  - XUL 内で使用する nsIJSON は Fx3 で実装済
- ワーカースレッド API
  - wk = new Worker("path/to/script.js"); で生成して  
wk.onmessage, wk.onerror, wk.postMessage を定義
  - まだ API の仕様は標準化されていない
  - バックグラウンドからは直接 DOM を操作できない
  - 重たい計算処理やオンラインに戻るときの処理などに



# TraceMonkey

- Firefox 3.1 の新 JavaScript エンジン
  - 目標は C や Java に匹敵する処理速度
  - 少ないメモリで高速な処理を実現するのが特徴
- 新しいタイプの JIT コンパイラを搭載
  - Tamarin (Tamarin-Tracing) の nanojit を移植
  - UC の Andreas Gal らの研究成果を反映
  - Tracing という手法を用いた JIT コンパイラ
  - Andreas Gal による Tracing の論文は 2006年～
  - TraceMonkey の開発は今年の 6月末～



# JavaScript = 遅い？



# JavaScript は何故遅い？

- JavaScript = 単純なスクリプトを処理する言語
  - そもそも Java があるのに JS ができたのはそのため
- 用途上そもそも高速化する必要性、需要がない
  - Web アプリケーションができたのはつい最近
- 開発者の数は C や Java に比べるまでもない
- 比較的新しい言語
  - Fortran: 1954~、C: 1972~、C++: 1982~、Java: 1990~、JavaScript: 1995~
- 誰もコンパイラの開発に力を入れていなかった



# 言語の速度

- 一昔前に C++ と Java のどちらが速いか議論された
  - 仮想マシンで動く Java は遅い！？
    - 逐次解釈では C の 20 倍以上遅かった
    - 中間言語への事前コンパイルで高速化
    - JIT によるネイティブコード生成で高速化
    - 現在では処理内容次第では Java の方が高速
  - パレートの法則
    - 処理の大部分は一部のコードに集中する
    - ボトルネックとなる部分を高速化すればよい



# JavaScript は何故遅い？

- 動的スクリプト言語
  - 演算前に型のチェックと変換が必要
- 配列が存在しない
  - Array は配列でなくハッシュオブジェクト(仕様)
    - array[3] は実際には array["3"] へのアクセス
    - すべてのプロパティアクセスでハッシュ計算が必要
- クラスが存在しない
  - プロトタイプベースのオブジェクト指向言語
  - 変数のクラスによる事前最適化が不可能



WEbAR.0

時は流れ…

PlAtf@xrm



高速化しなきや！



# JavaScript の高速化

- 動的スクリプト言語
  - ループ外で型チェックしてから固定型で演算
- 配列が存在しない
  - 配列として使われているなら配列として処理
  - 同じプロパティへのアクセスはキャッシュ
- クラスが存在しない
  - オブジェクトの定義を変更することはまれ
  - 同じ手順で定義したオブジェクトを同じクラスに
  - クラスチェックだけしてクラスに特化した処理

# Just In Time Compiler

- ボトルネックを高速なネイティブコードに変換
- 実行時に繰り返し処理される部分を検出
  - 既存の JIT は関数(メソッド)呼び出し回数をカウント
  - 一定回数以上呼び出されたメソッドをコンパイル
- 実行時情報を使った最適化が可能
  - 事前コンパイル型の C++ などより高速化可能
- Java は HotSpot という JIT の導入で大幅高速化
- Action Script 3 も JIT によって高速化
  - 但し AS3 の場合は静的型の導入が大きい



# Tracing JIT

1. ソースコードをバイトコード(中間コード)に変換
2. インタープリタでバイトコードを実行しつつ監視
  - 後方分岐発生時以外はペナルティ一切無し
3. 後方分岐のジャンプ先をループ始点候補として記録
  - ループ処理は必ず後方分岐を伴う
4. 一定回数以上ループしたら実行バイトコードを記録
  - 実際に処理するバイトコードだけを記録する
5. 記録された軌跡 (trace) をネイティブコードにコンパイル
6. ネイティブコードで高速処理



# Tracing JIT

```
var i,k=0;  
for (i=0; i < 1000; ++i) {  
    ++k;  
}  
alert(k);
```

トレース



A	iload_1	
	sipush 1000	
	if_icmpge B	
	iinc 2,1	
	iinc 1,1	
	goto A	後方分岐
B	getstatic System.out	
	iload_2	
	invokevirtual alert(int)	



# Tracing JIT

- ループのバイトコードをトレース
  - 全ての分岐結果を記録
  - 同一条件で実行される限りネイティブコードで処理
  - 記録した分岐結果と異なる場合は VM に処理を返す
  - switch 文なども分岐テーブルではなくトレース時に実行された特定条件との比較だけを行う
- ループ完了または特定条件発生時にトレース終了
  - 例外発生についてはトレース対象外
  - ネイティブメソッド呼び出しも基本的に対象外



# Tracing JIT

- ネイティブコードは既存の手法より高速化可能
- 実際処理される最小限のネイティブコードを生成
  - 既存の JIT では関数中の非実行部分もコンパイル
- ループ中の関数呼び出しもネイティブコードで最適化
  - 関数単位のネイティブコード生成では不可能
- 型についても当然ループ開始時のチェックのみに
- switch 文だったところも if 文として処理
  - トレース時と同じ分岐かどうかをチェックするだけ
- 同一条件ループでループのオーバーヘッドなし



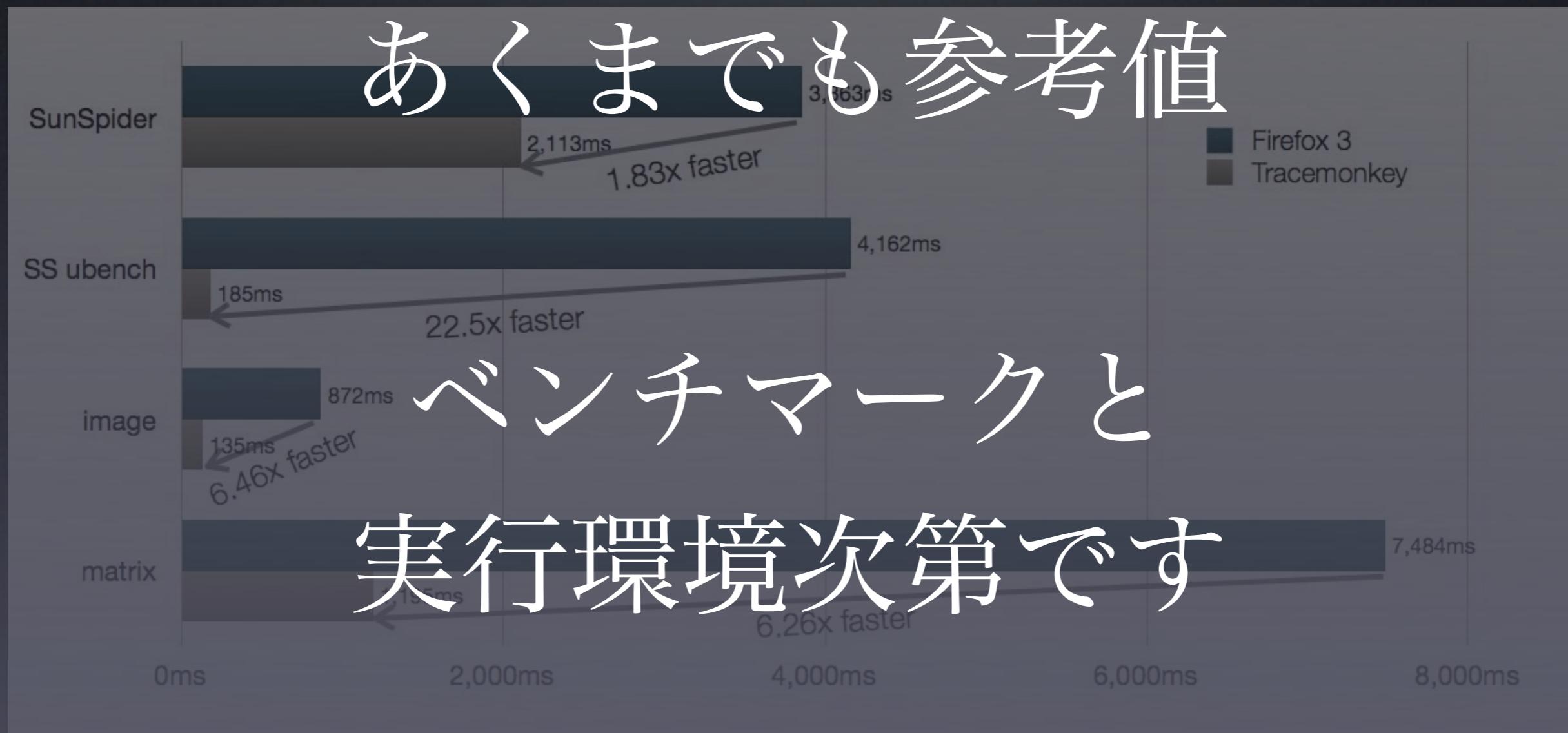
# Tracing the Web

- まだまだ開発は始まったばかり
  - ようやく安定してきた段階で高速化の余地はまだまだ
- 現在のトレース対象はネイティブ JS と一部の DOM
  - XPCOM のネイティブメソッド呼び出しがあると一部対応しているものを除きトレース対象外に
  - 純粋な JS オブジェクトの処理は非常に高速に
  - 画像変換など重たい数値計算処理には効果大
- 他言語から生成したバイトコードも処理
  - IronPython などの IronMonkey プロジェクトも…



# Fx3 vs TraceMonkey

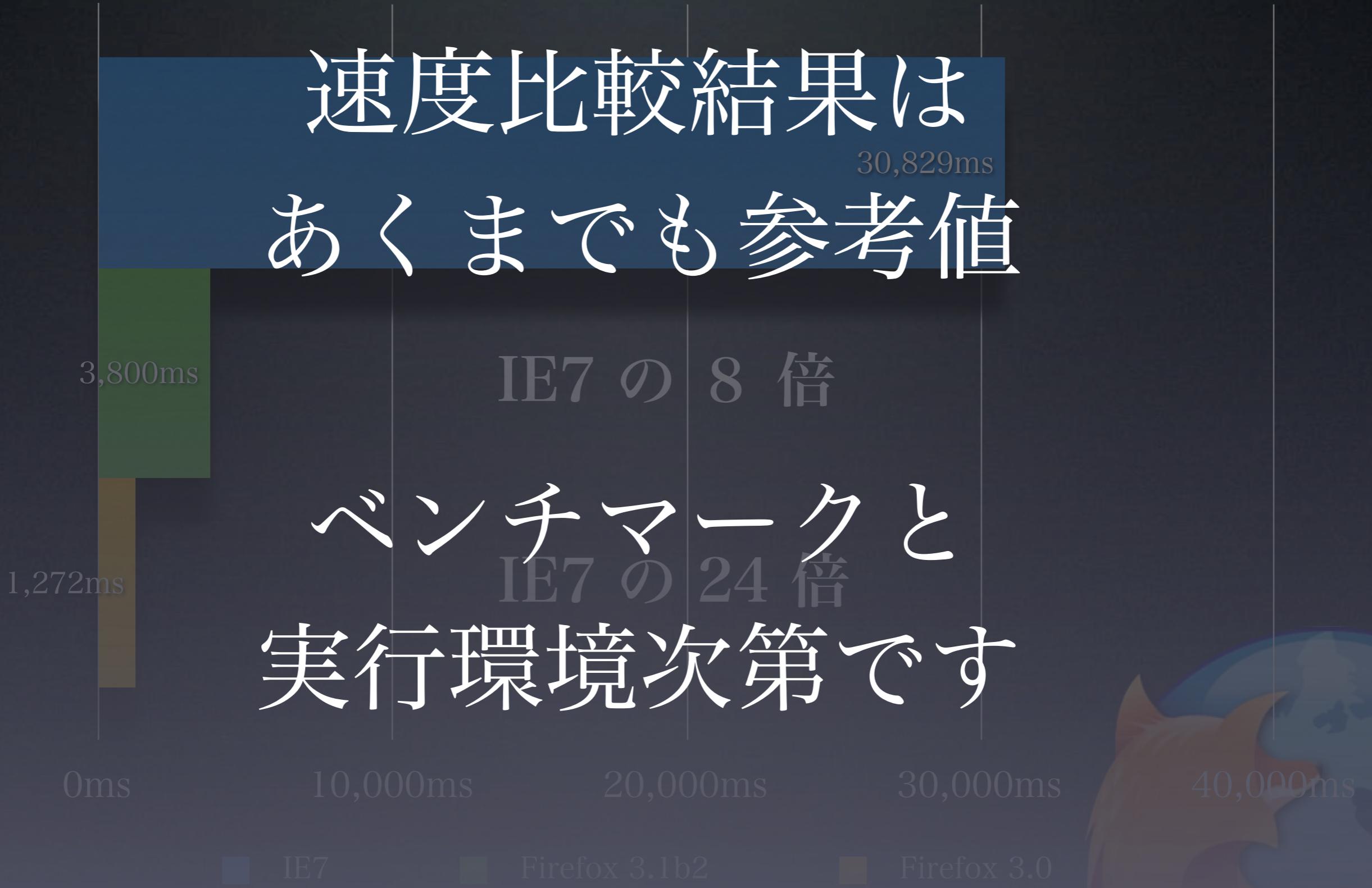
各種速度比較結果は月末)



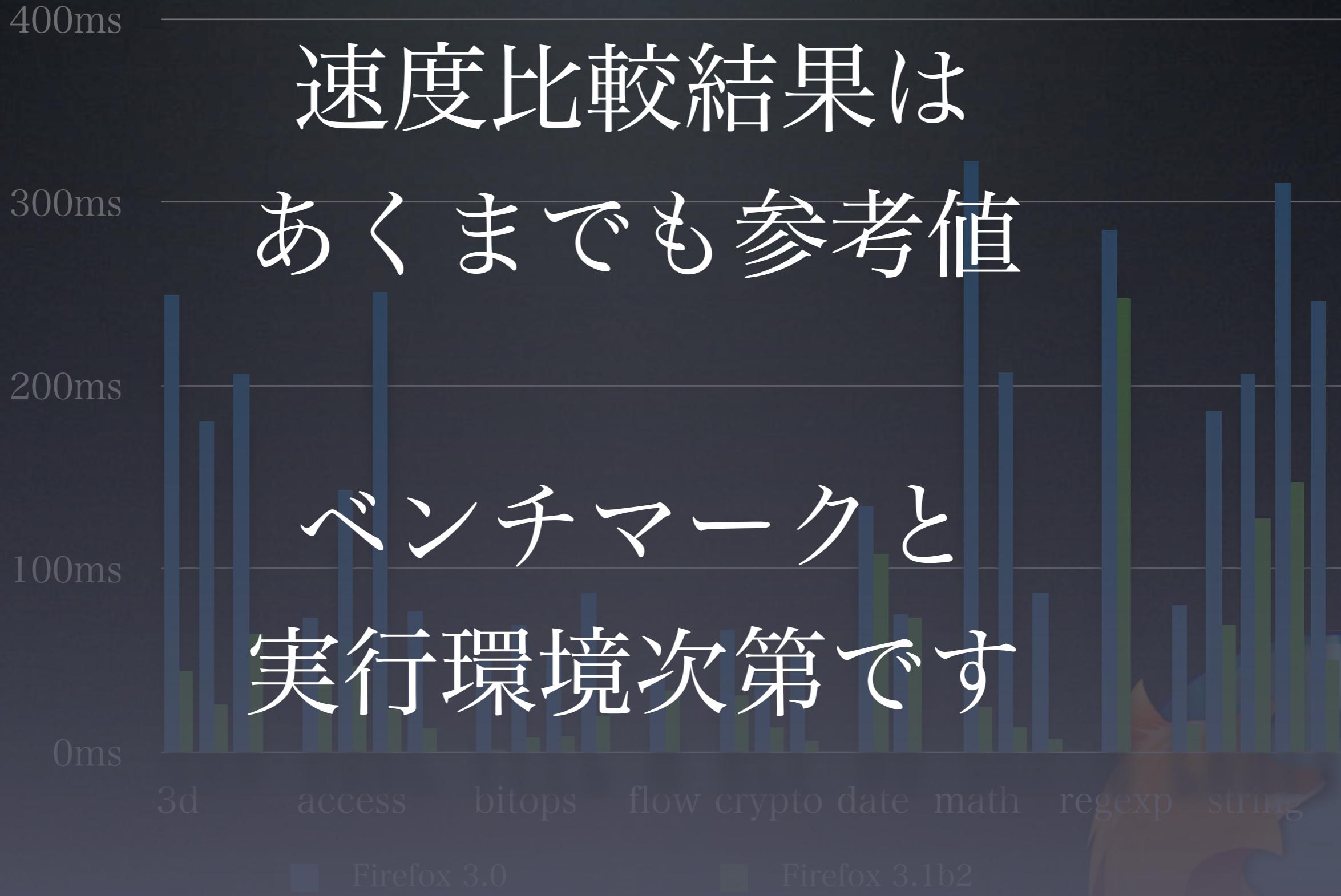
Benchmark by Brendan

[http://weblogs.mozilla.org/roadmap/archives/2008/08/tracemonkey\\_javascript\\_lightsp.html](http://weblogs.mozilla.org/roadmap/archives/2008/08/tracemonkey_javascript_lightsp.html)

# IE7 vs Fx3 vs Fx3.1b2



# Fx3 vs Fx3.1b2



# 補足: Tamarin との関係

- Tamarin は Adobe の Action Script エンジン
- Tamarin と SpiderMonkey の統合は中止
  - Tamarin は SpiderMonkey の高速化に追従できない
  - Tamarin には eval() など必要な機能が多数欠落
  - バイトコードや DOM との結合が強い
- TraceMonkey では Tamarin の JIT のみ移植
  - Tamarin-Tracing というブランチの JIT を移植
  - 将来的にはガベージコレクタなどの移植や統合も
- 相互のコードや成果を反映し合いながら開発



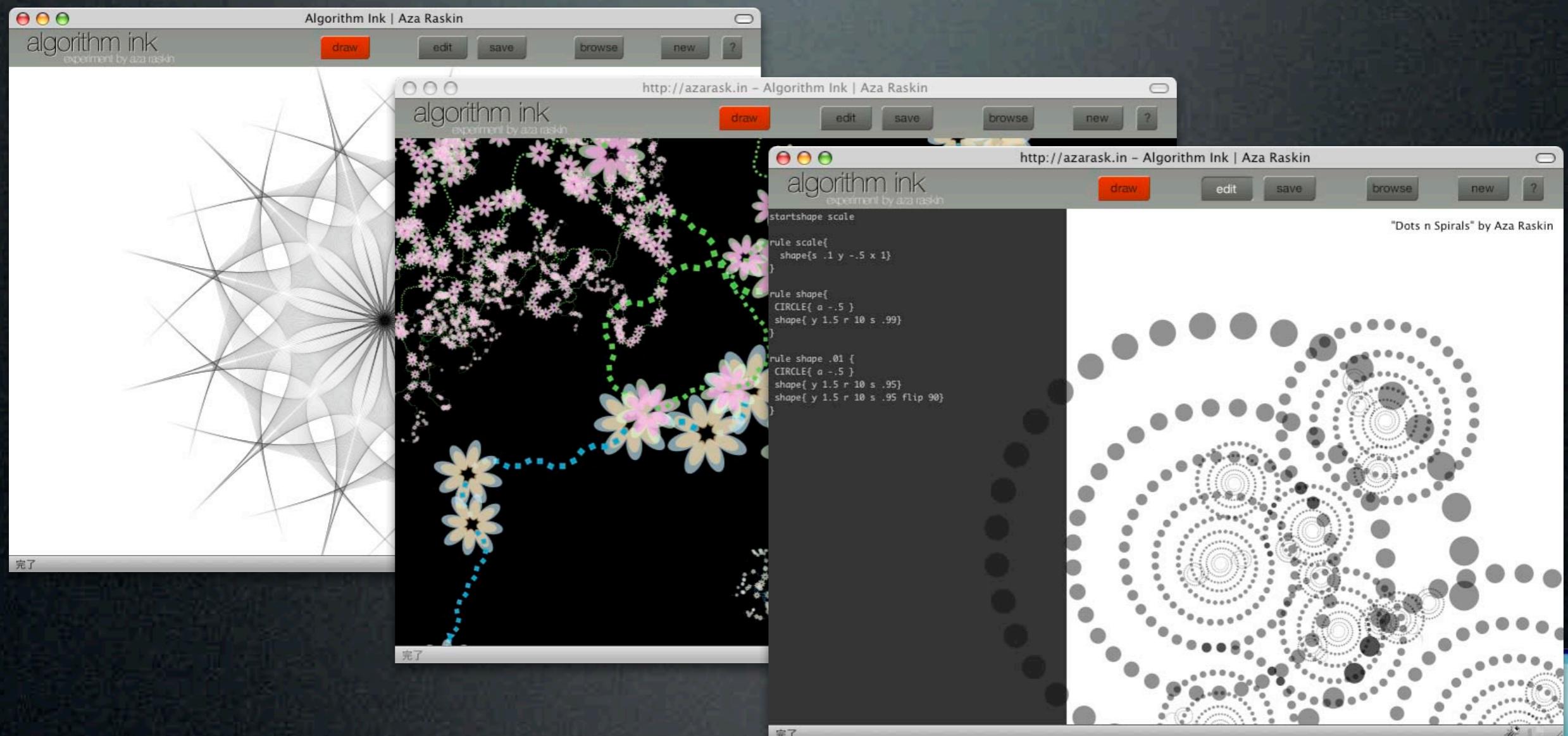


Photo by shiokuma  
<http://flickr.com/photos/shiokuma/631332559/>

# New Web



# Canvas + JavaScript

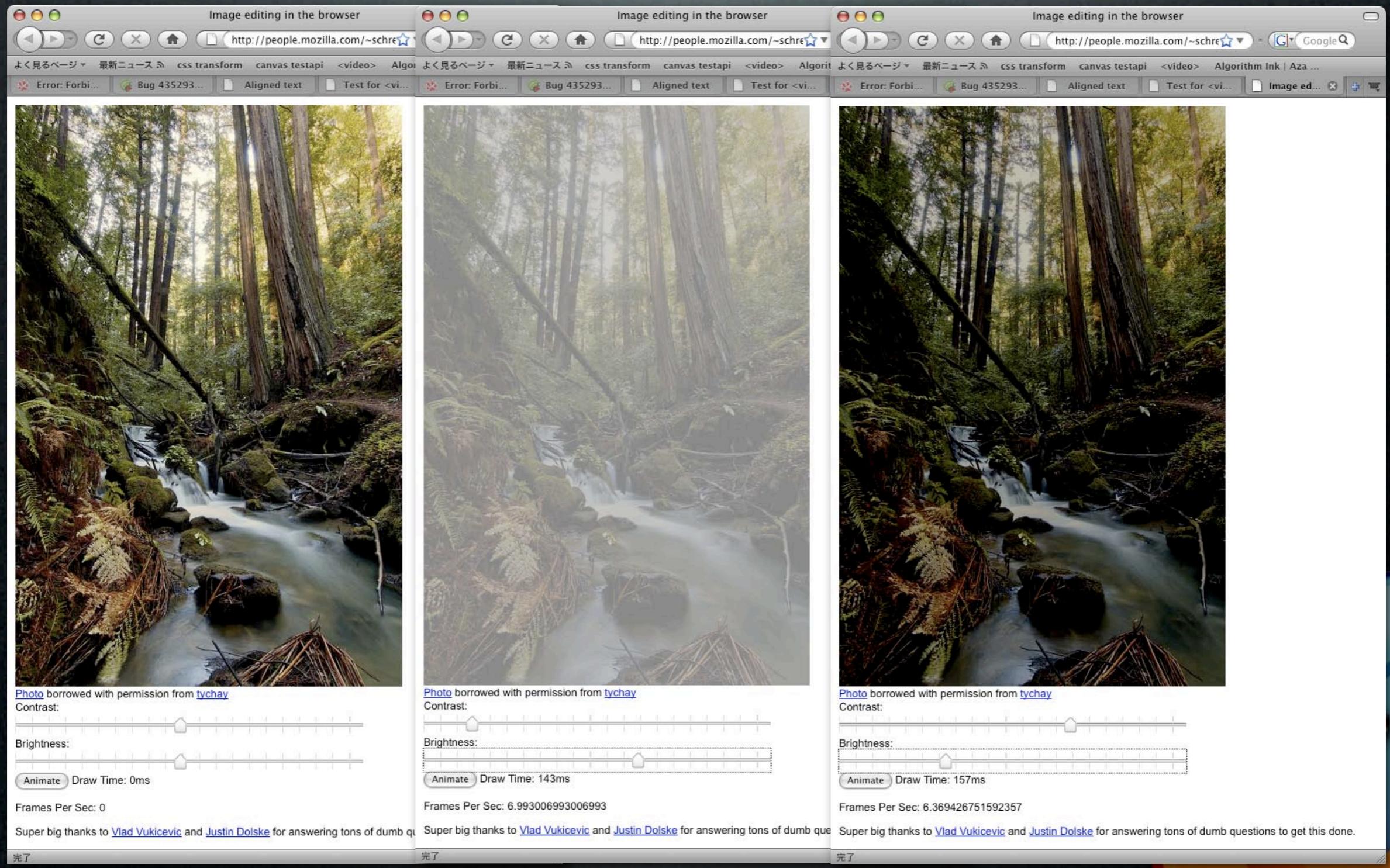


数行～数十行のコードでアート

<http://azarask.in/projects/algorithm-ink/>



# JavaScript で画像処理



Web = Platform





Photo by masahiko  
<http://www.flickr.com/photos/masahiko/986748017/>

# Quality Assurance

