




userChrome.js

～拡張機能の解体再構築～



## 本日のテーマ

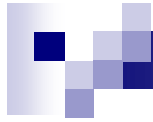
拡張機能を捨てる

(解体)



userChrome.js用スクリプト化する

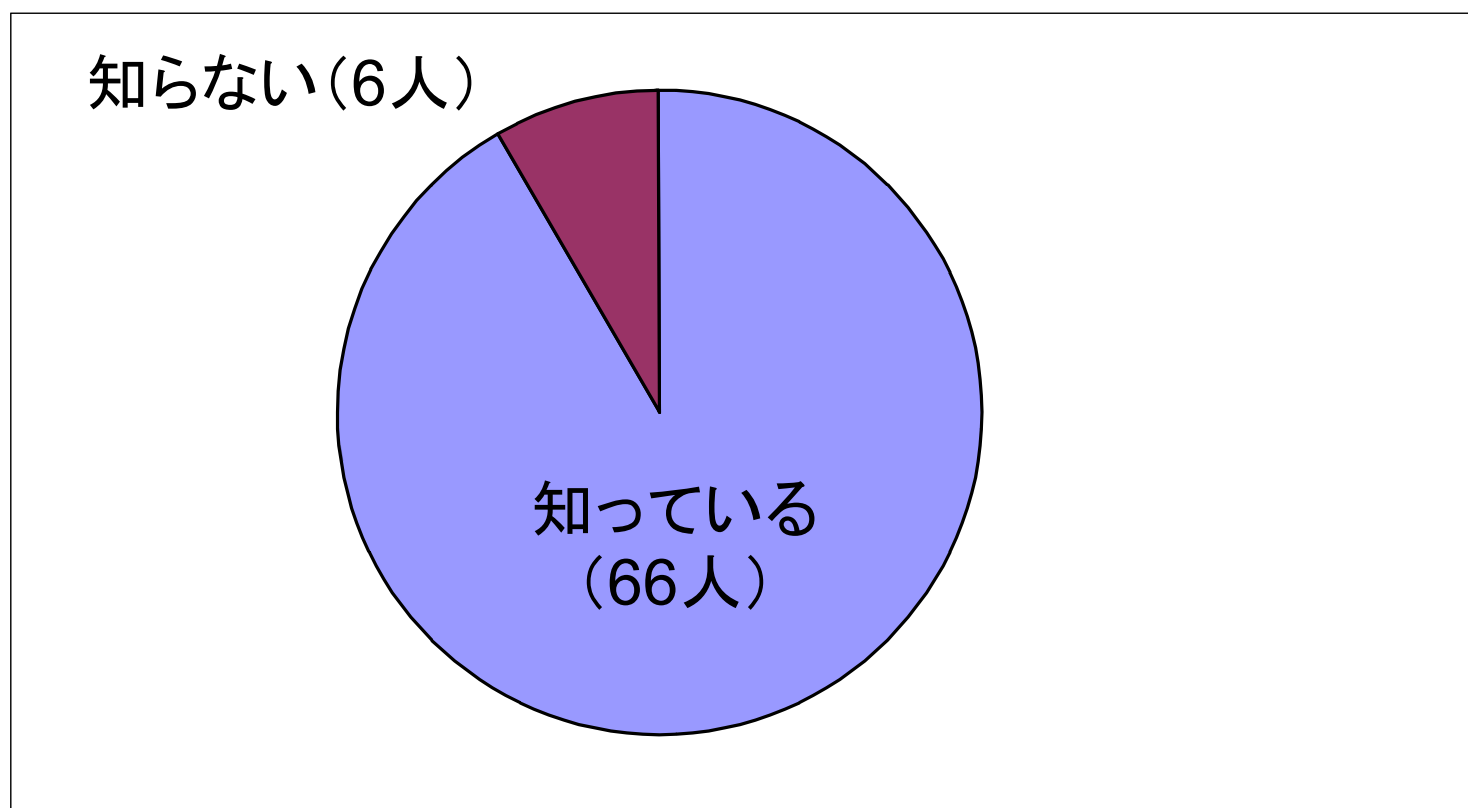
(再構築)



本題に入る前に...

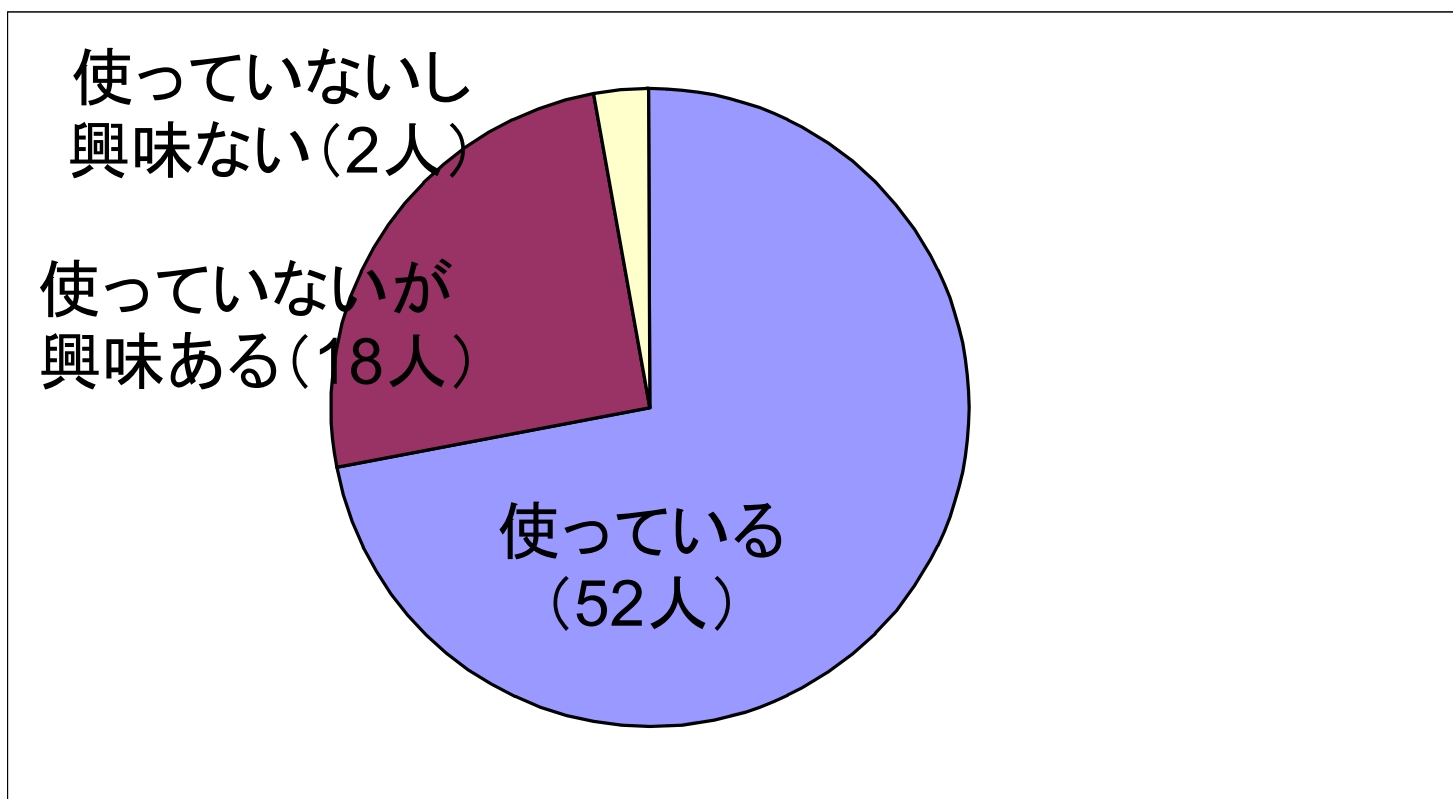
# 事前調査①

- 私のブログ訪問者72人に聞きました
- あなたはuserChrome.jsを知っていますか？



## 事前調査②

- 私のブログ訪問者72人に聞きました
- あなたはuserChrome.jsを使っていますか？



# 事前調査③

- みんなの拡張機能 ランキングによると...
- 603位 / 23ユーザ (少数派)

GOGA > みんなの拡張機能 > 拡張機能ランキング

みんなの拡張機能

■ 拡張機能ランキング

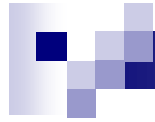
総投票数：8051

順位	拡張機能	グラフ	投票数	ステータス	コメント
601	<a href="#">OPML Support</a>		23	<input type="button" value="導入する"/>	<a href="#">コメント</a>
602	<a href="#">Bookmark All</a>		23	<input type="button" value="導入する"/>	<a href="#">コメント</a>
603	<a href="#">userChrome.js</a>		23	<input type="button" value="導入する"/>	<a href="#">コメント</a>



## まとめ(勝手な推測)

- userChrome.jsはごく一部のマニアに支持
- 一般ユーザーにはまだ浸透していない
- そもそも[addons.mozilla.org](https://addons.mozilla.org)へは未登録  
(配布先は[Mozillazine Forum](https://www.mozillazine.org))



# 本題

## userChrome.jsとは何か？






## userChrome.jsとは？①

- **userChrome.jsという名の拡張機能**  
(Firefox/Thunderbird/Seamonkeyなどに対応)
- userChrome.js自体の機能はきわめて単純
- **{プロファイルフォルダ}\chrome\userChrome.js**  
へ色々なスクリプトを記述することで、  
ブラウザウィンドウを開くと同時に読み込まれる
- **スクリプトによって多彩な機能を追加可能**



## userChrome.jsとは？②

- いわばXUL版Greasemonkey
- いわばJavaScript版userChrome.css
- 作者はzeniko氏(♂)




## userChrome.jsは何が売りか？

- ちょっとした拡張機能であれば、userChrome.js用スクリプトで代替可能
- つまり、拡張機能の数を減らして、Firefoxを軽量化できる
- スクリプトを直接編集して、とことん自分好みにカスタマイズできる



さっそく、userChrome.jsを  
導入してみよう



## 使い方①

- [userChrome.js 0.7](#)をインストール
- [Firefoxまとめサイト](#)から欲しいスクリプトを探す
- {プロファイルフォルダ}\chrome\userChrome.js  
へスクリプトを追記でコピー
- Firefoxの新しいウィンドウを開いて動作確認

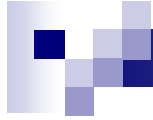
スクリプトの数が増えると、userChrome.jsが  
ごちゃごちゃになる。

そこで...

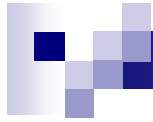
## 使い方②

- {プロファイルフォルダ}\chrome\userChrome.js  
へ UCJS Loader をコピー
- 探してきたスクリプトをコピーして、  
{プロファイルフォルダ}\chrome\\*\*\*\*\*.uc.js  
のようなファイル名で保存する

1スクリプト1ファイルとなるため、管理がしやすい  
UCJS Loaderは日本語(マルチバイト)にも対応  
ここから先は、UCJS Loaderの使用を前提とする



いったい、どんなスクリプト  
があるのか？

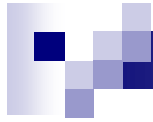


# ColorfulTabs.uc.js

by Gomita

- タブの色をカラフルにする

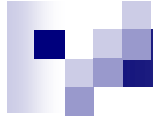




# CopyTabInfo.uc.js

by Gomita

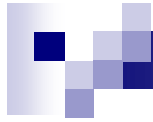
- タブを右クリックしてタイトルやURLをクリップボードへコピー



# VerticalToolbar.uc.js

by Gomita (未公開)

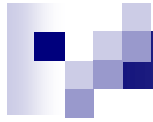
- 縦置きツールバーを追加
- ツールバーに置くボタンなどをカスタマイズするにはスクリプトを直接編集する



# MouseGestures.uc.js

by Gomita

- シンプルかつ軽量なマウスジェスチャ
- ジェスチャをカスタマイズするには  
スクリプトを直接編集する



# Drag'n'go.uc.js

by zeniko?

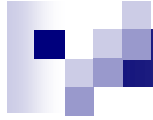
- シンプルなSuper DragAndGo
- リンクをドラッグドロップして新しいタブで開く



# TooltipTabPreview.uc.js

by Gomita (未公開)

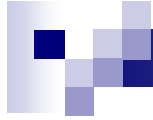
- タブのツールチップでサムネイルを表示
- Opera風



# open-url-with.uc.js

by nanto\_vi

- 右クリックメニューからページやリンクを Internet Explorer や Operaなどで開く



# youtubeDownloader.uc.js

by flyson

- YouTube動画のタイトル部分をクリックしてFLVファイルをダウンロード



# userChrome.js用 スクリプトを自分で作る



# 「再起動」メニューを追加するスクリプト

## ■ RestartMenu.uc.jsを作成し、エディタで編集

```
(function() {  
    // [再起動]メニューを生成  
    var restartMenu = document.createElement("menuitem");  
    restartMenu.setAttribute("label", "再起動");  
    // 再起動を実行する関数  
    var restartFunc = function() {  
        var startup = Cc["@mozilla.org/toolkit/app-startup;1"]  
            .getService(Ci.nsIAppStartup);  
        startup.quit(startup.eRestart | startup.eAttemptQuit);  
    };  
    restartMenu.addEventListener("command", restartFunc, false);  
    // [ファイル]→[終了]のひとつ上に[再起動]メニューを追加  
    var quitMenu = document.getElementById("menu_FileQuitItem");  
    quitMenu.parentNode.insertBefore(restartMenu, quitMenu);  
})();
```

## テクニック①：クローージャ

- スクリプト内で単純に「var hoge;」と変数作成すると、browser.xul内のグローバル変数とになってしまうことに注意
- そこで...
- スクリプト全体をクローージャにする

```
(function() {  
    // ここへ記述したスクリプトでは  
    // ローカル変数を使用可能  
    var hoge;  
})();
```

## テクニック② : E4XでXULオーバーレイ

- メニューやボタンを追加するためには、通常 `document.createElement()` を使用
- E4Xにより、スクリプト内に直接XULを記述してオーバーレイさせることが可能

```
var overlay =
    <overlay xmlns={ kXULNS }>
        <menupopup id="menu_FilePopup">
            <menuitem label="再起動"
                insertbefore="menu_FileQuitItem" />
        </menupopup>
    </overlay>;
overlay = "data:application/vnd.mozilla.xul+xml;charset=utf-8,"
    + encodeURIComponent(overlay.toXMLString());
document.loadOverlay(overlay, null);
```

## テクニック③ : E4Xでスタイルシート追加


- E4Xにより、スクリプト内に直接CSSを記述してスタイルシートを追加することが可能

```
var style = <![CDATA[
    .tabbrowser-tab[selected="true"] > * {
        background-color: gold !important;
    }
]]>.toString().replace(/¥s+/g, " ");
var sspi = document.createProcessingInstruction(
    'xml-stylesheet',
    'type="text/css" href="data:text/css,' + encodeURIComponent(style) + '"
);
document.insertBefore(sspi, document.documentElement);
sspi.getAttribute = function(name) {
    return document.documentElement.getAttribute(name);
};
```

## テクニック④ : data:URLで画像埋め込み

- data:URLにより、アイコン画像などのデータをスクリプトへ埋め込み可能

```
restartMenu.setAttribute("class", "menuitem-iconic");  
restartMenu.setAttribute("style", "list-style-image:  
url (data:image/png;base64, iVBORwOKGgoAAAANSUUhEUgAAABAAAAAQAYAAAF8%2F9hAAA  
ABGdBTUEAAK%2FI NwWK6QAAABIORVhOU29mdHdhcmUAQWRvYmUgSW1hZ2VSZWFkeXHJZTwAAAHs  
SURBVDjLtZPpTlpRFIV5Dt7A0ESr1kYNThGnSomI ihPoNVi5Qp3RgBgvEERpRW1BRBAcMEDUtIk  
djKk4otK0Jdr2vgxZ3kA0MYoaG3%2BcX2evb529zt4sAKz%2F0awnASgCBNm5LaE7vjVDutkA4m  
MdLV4TkvcCuvba2lqd1pDhWA33mQU%2B2oXVv07YfpoxuNWFuqVXoeqFCnZcgJwRm04p%2BGk3F  
s9t8PyZx%2FK5Hfbf03CGLRj62g2%2BrSR0KOD%2BvZXUB1Xw%2Fou5usJWjAaU0Gz3w%2FrjHe  
y%2FZjDLvKTD34KSyXzyBkC2JaYd4feMqyNa30QTREqePI Xjr qSq5ssj5hMjTmd66ALDKDLm0jc  
A0s%2BNID6JIFmvQaNXANEKX3l5x7NyqTcb7Zg8GYtC0LoXuPcbha6XV0VlU4WUzE9gPKjF2CGF  
bE3G3QAmafDnShETF3iKTZyIblcnZa4Syi%2FdeD6USscFCJwV6Fwn8NonQak5Hy1L9TAcjkJ%2  
FoAG1p0a1hYdnfcnkRQCBoxyyNYLp1YCJoB7GlwqGgxGod%2FoZsQoNDiHSepNCceeAN8uF1CvG  
xJE25rofoc%2B3bIKPqQ2VUnKxIYN85yty3eWh216LeKUTOScayVGI IH0g5S%2B1JJB%2B8CxxTl  
rWkH7WNTNIPAlwA9Gm70cXUHxUAAAAASUVORK5CYII%3D)");
```



ところで、最近userChrome.js  
の新しいバージョンが出ました



## userChrome.js 0.7から0.8へ

- つい先日、[userChrome.js 0.8](#)がリリース
- あらゆるchromeウィンドウへフック可能
- ブラウザウィンドウだけでなく、  
[ブックマークサイドバー](#)や  
[アドオンマネージャ](#)などへの機能追加も可能
- 内部処理は独自XPCOM
  - “domwindowopened”通知を監視して...
  - “load”イベントへリスナを追加する

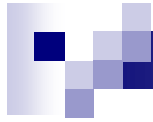


userChrome.js 0.8を0.7同様に使うには？

- userChrome.jsの先頭へ以下の内容を追加
- ブラウザウィンドウ以外でのスクリプト読み込みを中止させる

```
if (location != "chrome://browser/content/browser.xul")  
    throw "stop";
```






# userChrome.jsの 今後の展望



# userChrome.jsの今後の展望①

- 事前調査で多く寄せられた意見
  - Stylishのような管理機能があると便利ですね.
  - Stylishのような管理機能があれば使う。
  - StylishみたいなGUIが付いたりすると良いと思います.
  - スクリプトを管理する何かがあるといい。
- ちなみに、Stylishとは、userChrome.css管理用の拡張機能である
- つまり...

- 
- userChrome.jsスクリプト管理用の拡張機能が今もっとも望まれている
  - こんな機能(案)
    - スクリプトの一覧表示
    - スクリプトのインストール／削除
    - スクリプトの有効／無効
    - スクリプトの編集
    - スクリプトの更新(バージョンチェック)



## userChrome.jsの今後の展望②

- 現状あちこちのWebサイトにスクリプトが散在
- 目的のスクリプトを探し出すのに一苦勞
- そこで...
- 「[userscripts.org](http://userscripts.org)」(Greasemonkey用)や「[userstyles.org](http://userstyles.org)」(Stylish用)のように...  
「[userchromejs.org](http://userchromejs.org)」があるとうれしい

## userChrome.jsの今後の展望③

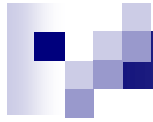
- メタデータ記法の統一
- スクリプト内にコメントでメタデータを埋め込む
- 下記はGreasemonkey風の案

```
// ==UserScript==
// @name      Restart Menu
// @author    Gomita
// @description Add [Restart] menu under [File] menu
// @overlay    chrome://browser/content/browser.xul
// @version    0.1.2
// ==/UserScript==
(function(){
    ...
})();
```



## userChrome.jsの今後の展望(まとめ)

- まとめると...
- スクリプト管理用拡張機能  
+  
スクリプト配布サイト  
+  
メタデータ記法の統一
- 三者の合わせ技によって、userChrome.jsが  
より多くのユーザへ広まること間違いなし



ちよつと待った！



## userChrome.jsの危険性

- userChrome.jsのスクリプトは「**chrome権限**」で動作する
- 拡張機能同様にXPCOM使い放題  
つまり、**何でもやり放題**
- 見知らぬサイトのスクリプトをそのままコピーして使うのは危険
- userChrome.jsがaddons.mozilla.org未登録（意図的に登録してない？）理由はこの辺が大きいかも





## 最後に

- これから拡張機能開発に手を出そうと思っている方は、まずuserChrome.js用スクリプトの作成や改造から始めてみるのもありかも
- userChrome.jsによってXULやJavaScriptの技術の裾野が広がることに期待したい