

Crossweight

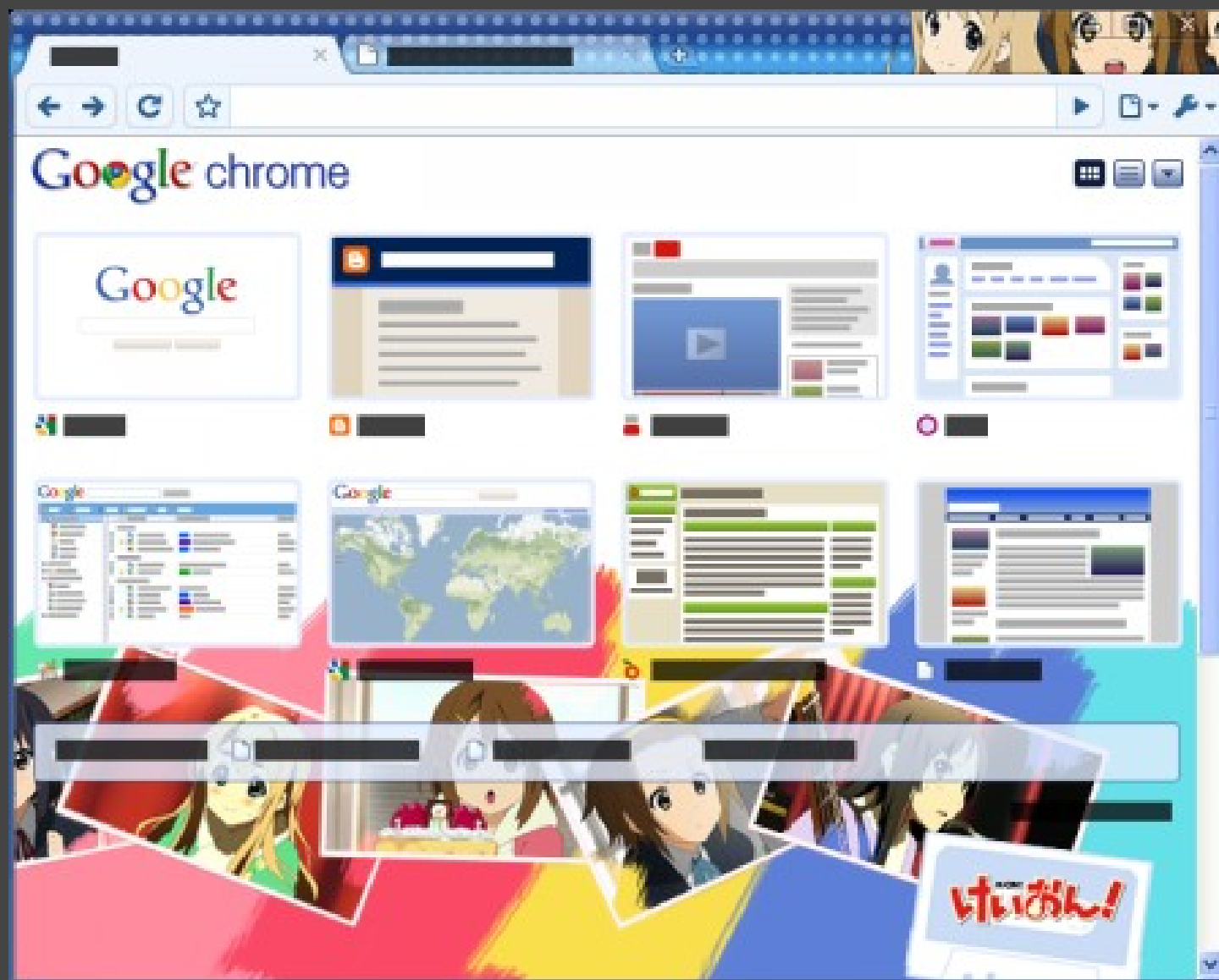
Crossweight とは
Google Chrome 用のテーマを
Firefox 3.6 で動かすアドオンパッケージ

Chrome Theme の正体

Chrome Theme は JSON 形式の設定ファイルと
壁紙用の画像ファイルを Zip でパッケージしたもの

つまり

Firefox 3.6 の Lightweight Theme と似た形式



テーマを移植してみた

移植方法

1. Chrome Theme を解凍し Crossweight アドオンの content 下にコピー
2. Window の Load 時に Ajax で設定ファイルを読み込む
3. 読み込んだ設定ファイルは Firefox の LwTheme 形式に変換される (共に JSON 形式)
4. LightweightThemeManager.jsm を使用してインポートする

Chrome Theme と Lightweight Theme 設定

name = id

name = name

headerURL = theme.images.theme_frame

textcolor = theme.colors.bookmark_text

accentcolor = theme.colors.frame

(Lightweight Theme = Chrome Theme)

Tips

header 画像に theme_toolbar ではなく theme_frame を使用したのは Chrome Theme の大半が theme_frame にキャラクター等のアクセントをくわえているため。

Chrome Theme の色設定は RGB 形式を使用しているので Hex 形式に一度変換する必要がある。



いっけん移植できているように見えるが...

Lightweight Theme には repeat 設定がされていない。

そのため repeat される事を前提とした Chrome Theme を移植する場合には Javascript を使った調整が必要になる。

そこで Window の CSS に background-repeat: repeat-x; を追加した。

```
$e("#main-window").style("backgroundRepeat","repeat-x");
```



ウィンドウを最大化した状態

てか、肝心の軽音部員いなくね？

軽音部員の画像は別ファイル
(theme_frame_overlay) になっていて
背景画像を合成しているらしい



そこで Firefox 3.6 の新機能
Multiple background で再現してみた

```
background-image: url(画像のURL), url(画像のURL);
```



今度こそ完成？

どうせなので about:blunk も再現

問題

about:blank から chrome 下の画像へアクセスできない。

解決方法

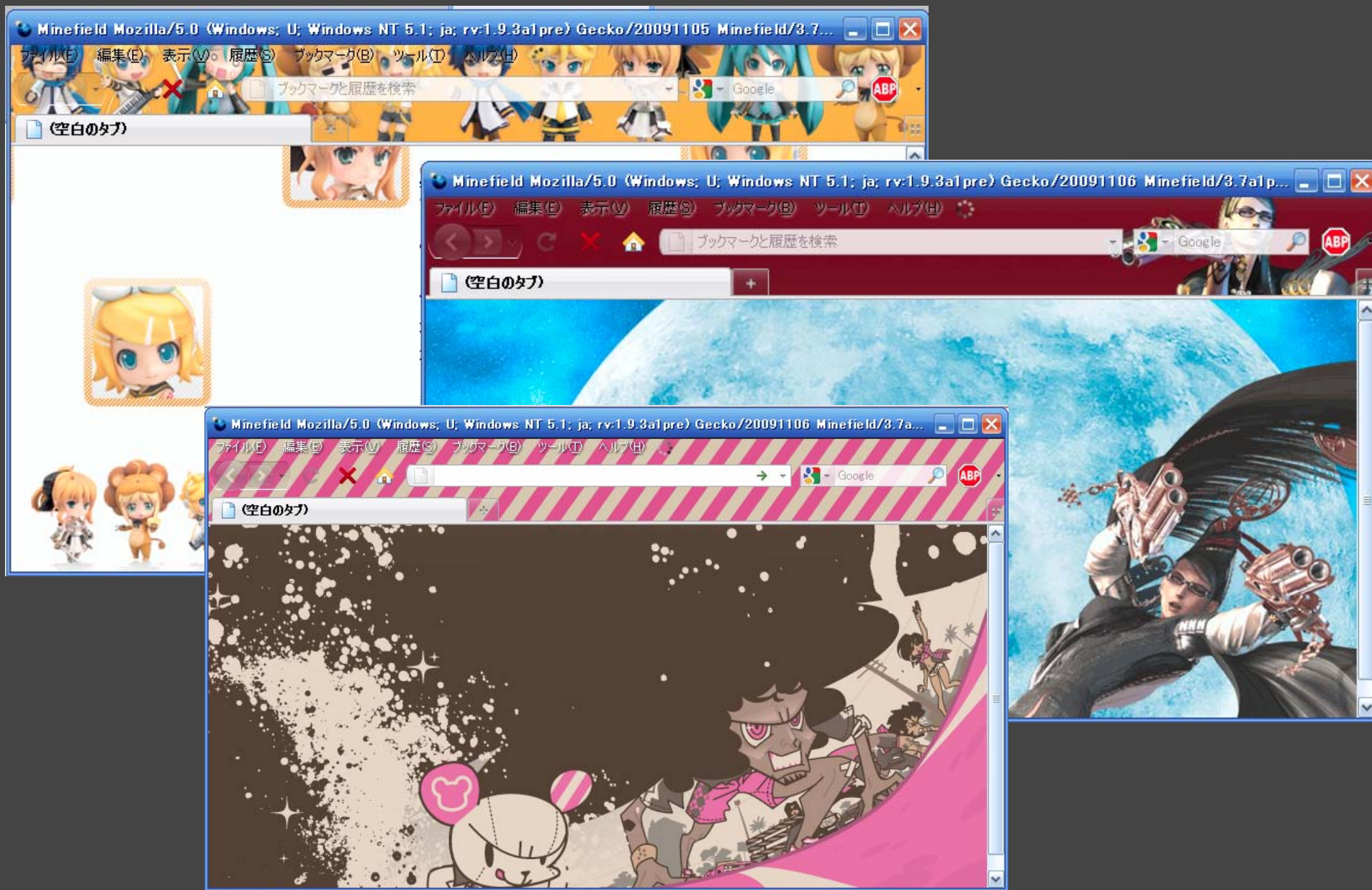
そこで一度画像を canvas に描画し Base64 形式に変換

data スキームで画像を表示する事でプロトコルによるセキュリティを回避

```
var img = new Image();
img.onload = function(){
var canvas = document.createElementNS("http://www.w3.org/1999/xhtml","canvas");
var ctx = canvas.getContext('2d');
ctx.drawImage(img, 0, 0, img.width, img.height);
var data=canvas.toDataURL("image/png");
body.style.backgroundImage="url("+data+)";
}
img.src = // 画像のアドレス
```



about:blunk もけいおん！ にしてみた



他のテーマの移植例