

Coder's High 2017

2017.11.4

CSS Nite LP54「Coder's High 2017」

コーダーの前仕事、後仕事

イー・ネットワークス 前川 昌幸

今回の内容

コーディングのワークフローの流れで発生する各種作業のうち、実際にコードを書く「前」と「後」の作業についてのTipsやツールを紹介

私の属性

- サーバー/ネットワークエンジニアからデザイナーまで
在籍している社員20人弱の地方の制作会社
- 仕事はほぼ受託制作での実装
- デザイン事務所や代理店からの依頼
- 東京:地方 = 7:3

アジェンダ

- 前作業
 - 仕様の確認
 - 作業環境の構築
 - 画像の書き出し
- 後作業
 - デザイン反映チェック
 - エミュレーター
 - トラッキングコード

前作業

仕様の確認

対象ブラウザの確認

「いまさら?」と思わないように

Internet Explorer

Microsoftのサポートポリシー上、

現行利用できるのはInternet Explorer 11のみ
(Windows Serverは除く)

<https://www.microsoft.com/ja-jp/windowsforbusiness/end-of-ie-support>

最新版ではない Internet Explorer のサポートが終了

2016年1月12日より、各オペレーティングシステムにおける最新版の Internet Explorer のみがテクニカル サポートとセキュリティ更新プログラムを受けられるようになりました。Internet Explorer 11 は最終版の Internet Explorer であり、Windows 7、Windows 8.1、および Windows 10 で、セキュリティ更新プログラム、互換性の修正、およびテクニカル サポートを引き続き受けることができます。

[今すぐアップグレード >](#)



Android

2015年、主なキャリアで発売されたAndroid端末のバージョンは4.4以降

2015年1月に4.3以前のWebView（標準ブラウザ）へのセキュリティパッチの提供を打ち切り

https://plus.google.com/+AdrianLudwig/posts/1md7ruEwBLF

Google+ ログイン

おすすめアカウント

Google+ に登録

問題を報告

ヘルプ

 **Adrian Ludwig** ▶ 一般公開 2015/01/24

Following public discussion of vulnerabilities in versions of Webkit last week, I've had a number of people ask questions about security of browsers and WebView on Android 4.3 (Jellybean) and earlier. I want to provide an update on what we're doing and guidance on steps that users and developers can take to be safe, even if your device is not yet running Lollipop.

Keeping software up to date is one of the greatest challenges in security. Google invests heavily in making sure Android and Chrome are as safe as possible and doing so requires that they be updated very frequently. With Google's assistance, Android device manufacturers (OEMs) have been moving rapidly to improve the rate that devices are updated and to ship devices with the most recent versions of Android. We provide patches for the current branch of Android in the **Android Open Source Project (AOSP)** [<https://source.android.com/>] and directly provide Android partners with patches for at least the last two major versions of the operating system.

Improving WebView and browser security is one of the areas where we've made the greatest progress. Android 4.4 (KitKat) allows OEMs to quickly deliver binary updates of WebView provided by Google, and in Android 5.0 (Lollipop), Google delivers these updates directly via Google Play, so OEMs won't need to do anything. Until recently we have also provided backports for the version of WebKit that is used by Webview on Android 4.3 and earlier. But WebKit alone is over 5 million lines of code and hundreds of developers are adding thousands of new commits every month, so in some instances applying vulnerability patches to a 2+ year old branch of WebKit required changes to significant portions of the code and was no longer practical to do safely. With the advances in Android 4.4, the number of users that are potentially affected by legacy WebKit security issues is shrinking every day as more and more people upgrade or get new devices.

There are also steps users and developers can take to mitigate the risk of potential exploitation of WebKit vulnerabilities without updating to Lollipop. Using a browser that is updated through Google Play and using applications that follow security best practices by only loading content from trusted sources into WebView will help protect users.

When browsing on any platform, you should make sure to use a browser that provides its own content renderer and is regularly updated. For instance on Android, **Chrome** [<http://goo.gl/eISkZX>] or **Firefox** [<http://goo.gl/Q5X6e3>] are both great options since they are securely updated through Google Play often: Chrome is supported on Android 4.0 and greater, Firefox supports Android 2.3 and greater. Chrome has been the default browser for all Nexus and Google Play edition devices since 2012 and is pre-installed on many other popular devices (including Galaxy devices from Samsung, the G series from LG, the HTC One series, and the Motorola X and G), so you may already be using it.

この半年間で実際にあった対応

- Internet Explorer 8
- Android 4.0

事前に仕様を確認して、
Flexboxの記述や、
モバイルスタートの
レスポンシブWebデザインで大丈夫かを
確認しておく

その他確認項目

- ファイル命名規則（「@」がダメ、など）
- ファイルパスの記述ルール（相対パス/絶対パス/ルートパス）
- JavaScriptライブラリ（jQueryのバージョンなど）

作業環境の構築

いわゆるタスクランナーなどの
ツールを利用するときに…

「黒い画面」が
苦手という人は多い

bat/commandファイルの活用

.batファイル(Windows)/.commandファイル(macOS)に
コマンドを記述することで、ダブルクリックで起動することが可能

※macOSでは実行権限を付与する作業が必要

start.command — codeset

```
1 # macOSでは必要な行
2 cd `dirname $0`
3
4 #Windowsのbatファイルでは下の行だけでOK
5 npm start
```

- FOLDERS
- codeset
 - _styleguide
 - htdocs
 - node_modules
 - src
 - .editorconfig
 - .node-version
 - /* gulpfile.js
 - npm-debug.log
 - /* package.json
 - start.command
 - /* webpack.config.js
 - yarn.lock

codeset

よく使う項目

- maekawa
- Dropbox
- スクリーンショット
- work
- AirDrop
- アプリケーション
- デスクトップ
- 書類
- ダウンロード
- Creative Cloud...
- OneDrive - (株)...

デバイス

- リモートディスク

codeset

- node_modules
- README.md
- _styleguide
- gulpfile.js
- htdocs
- node_modules
- npm-debug.log
- package.json
- src
- start.command
- webpack.config.js
- yarn.lock

```
# macOSでは必要な行
cd `dirname $0`

#Windowsのbatファイルでは下の行だけでOK
npm start
```

start.command

105 バイト

作成日 今日 22:08

変更日 今日 22:09

最後に開いた日 今日 22:15

[タグを追加...](#)

Line 4, Column 27

bat/commandファイルの活用

弊社では、

- 01-project-start.bat(npm install)
- 02-watch_and_build.bat(gulp)

を作業ファイルの一式に入れています

node.jsの管理

- 先日、2017年10月30日にnode.jsのLTSが6.11から8.9に
- パッケージに動くもの、動かないものがでるかも？
- 作業者間でのバージョンを揃える

nodenv/nodist

- nodenv(macOS)/nodist(Windows)で管理
- .node-versionファイルを共有
- しばらく期間が経たあとの改修などにも

<https://www.to-r.net/media/anyenv/>

to-R Media



Tools node.js 環境構築

anyenvとnodenvでNode.jsのバージョン管理をしよう！

2017.10.02

中村 勇希

いいね! 6 シェア ツイート はてなブックマーク あとで読む



Node.jsは開発環境によって異なるバージョンを求められることがあります。この記事ではNode.jsに限らず他の言語でもバージョン管理ができる「anyenv」を使った環境構築を行います。

この記事を取り扱った内容は以下の通りです。

https://qiita.com/satoyan419/items/56e0b5f35912b9374305

Qiita キーワードを入力 Release Qiitaユーザー向けのMastodon、Qiitadon を試験的に公開しました ユーザ登録 ログイン

nodistでNode.jsをバージョン管理

121 いいね | 0 コメント

いいね

Node.js 4828 | npm 799 | nodist 21

satoyan419 2017年10月13日に更新 4

ストック

※この記事はWindowsを前提として書かれています。
※Macの場合は、「[nodebrewでNode.jsをバージョン管理](#)」を参照してください。

Node.jsのインストール



nodistというツールを使ってNode.jsのインストールやバージョン管理をおこないます。

nodistのインストール

nodistのページからインストーラーをダウンロードして、nodistをインストールします（下図参照）。以前はシステム環境変数の設定が必要だったようですが、現在ではインストーラーがすべて設定してくれます¹。

Latest release

v0.8.8

e4c1326

marcelklehr released this on 17 Dec 2016 · 5 commits to master since this release

v0.8.8

Tweet B! 25 G+ Like 1 Pocket 46

satoyan419 201 Contribution フォロー

人気の投稿

- nodistでNode.jsをバージョン管理
- nodebrewでNode.jsをバージョン管理 - Node.jsとRubyの環境構築(2)
- コーディング作業を加速する Sublime Text 活用法
- rbenvでRubyをバージョン管理 - Node.jsとRubyの環境構築(3)
- Homebrewをインストール - Node.jsとRubyの環境構築(1)

ローカルサーバー

構築中、あるかないかで作業効率は大きく変わる

Browsersync

- タスクランナーを活用するひとにはおなじみ
- 単独でも活用可能

https://qiita.com/ta-ke-no-bu/items/e1972ff0f2ddfaf84b2b

Qiita キーワードを入力 Release Qiitaユーザー向けのMastodon、Qiitadon を試験的に公開しました ユーザ登録 ログイン

はじめてのbrowserSync

0 いいね | 0 コメント

👍 いいね

📦 BrowserSync 38 📦 Node.js 4828 📦 npm 799 📦 新人プログラマ応援 867

📁 ta-ke-no-bu 2017年10月16日に更新 🔄 3 📁 ストック

はじめて使ってみて社内共有用に書いたやつを転載

目次

- browserSyncについて
- できること
- 5分でできる簡易インストール
- 管理画面の設定
- まとめ

browserSyncについて

Live Reloadができるnode.jsパッケージ

[公式サイト](#)

Adobe Edge Inspectが似たようなことができるらしいけどbrowserSyncはPCでインストールするだけで拡張機能のインストールや端末へのアプリインストールも不要

参考：[Adobe Edge Inspect](#)

同期中の動作のみ

リモートデバッグ、コード調査イメージ (画像はAdobe Edge Inspectのもの)

[adobeedge-8-638.jpg](#)

できること

「福島県」【一般内科】外来・訪問診療・人工…
【求人の特徴】一般内科・透析内科医師の募集
年収：1300万円～ [求人詳細](#)

「千葉県」【小児科】小児科・外来のみ
【求人の特徴】2017年9月千葉駅直結オー…
年収：1500万円～ [求人詳細](#)

「兵庫県」【一般内科】病棟管理・外来診療の…
【求人の特徴】【人気の神戸市内】負担の少…
年収：1600万円～ [求人詳細](#)

医師の転職情報サイト
Dr.転職ナビ

[Tweet](#) [B! 0](#) [G+](#) [Like 0](#)

[Pocket](#) 1

 **ta-ke-no-bu**
106 Contribution
[フォロー](#)

人気の投稿

- web制作が捗りそうな便利なwebツール
【随時更新】
- webpack2.2.1にriotjs-loader4.0.0を追加して実行したらエラーが出る
- 【疑似要素をjsで動的に】幅100%のサブメニューを設置しホバーで吹き出しつける

今からはじめる英会話
身近なGOALは3月!
自分に必要な
英語回路 🔍 をつくる。

Browsersyncの弱点

- PHPが動作しない
- UTF-8以外の文字コードでは文字化け

ローカルのApache環境

- XAMPP/MAMPが有名

The screenshot shows the XAMPP website interface. At the top, there is a navigation bar with links for Apache Friends, Downloads, Add-ons, Hosting, Community, and About, along with a search bar and a language selector set to JP. The main heading reads "XAMPP Apache + MariaDB + PHP + Perl". Below this, a large blue banner displays the XAMPP logo (an elephant) and a large checkmark icon, with the text "ようこそ! MAMP のインストールに成功しました。" (Welcome! MAMP installation successful). To the right, there are buttons for "Linux 向け XAMPP 7.1.10 (PHP 7.1.10)" and "OS X 向け XAMPP XAMPP-VM (PHP 7.1.10)".



Linux 向け XAMPP
7.1.10 (PHP 7.1.10)

OS X 向け XAMPP
XAMPP-VM (PHP 7.1.10)

M for OS X available!
Try it now

PHP

PHPの設定を知るには、[phpinfo](#) の出力内容を見てください。

MySQL

MySQLデータベースは [phpMyAdmin](#) を使って管理できます。

スクリプトからMySQLサーバに接続するには、以下の設定を使用してください。

ホスト localhost

MAMP のバージョン

4.2 → 最新バージョンがインストールされています。

ニュース

[Mobile Testing with MAMP Viewer](#)

XAMPP/MAMPの場合

複数の案件が並行した場合、やや手数が必要

- DocumentRootの入れ替え
- confファイルの編集
- hostsファイルの編集

など

VirtualHostX

<https://clickontyler.com/virtualhostx/>

macOSのみ

- GUIでホスト名やconfを編集
- xip.ioを利用したLAN内でのアクセス
- \$49

VirtualHostX

Virtual Machine is running

sample

Job

Domain Name: sample .vhx.host

Local Path: /Users/maekawa/work/markup-code-set/codeset/htdocs Choose...

Web Root: **ドキュメントルートになるフォルダ**

Enable this website

Auto Refresh... ?

HTTPS...

http://sample.192.168.95.1.xip.io ?

Custom Directives

```
<VirtualHost>  
  
</VirtualHost>
```

```
<Directory>  
AddType application/x-httpd-php .html  
  
</Directory>
```

ホスト名を指定

**同一のLAN内
でアクセス可能
(Port:8080)**

**apacheの設定も
記述可能**

TEST

PHP Version 5.5.9-1ubuntu4.21 

System	Linux vagrant-ubuntu-trusty-64 3.13.0-113-generic #160-Ubuntu SMP Thu Mar 9 09:27:29 UTC 2017 x86_64
Build Date	Feb 9 2017 20:54:17
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/05-opcache.ini, /etc/php5/apache2/conf.d/10-pdo.ini, /etc/php5/apache2/conf.d/20-json.ini, /etc/php5/apache2/conf.d/20-mysql.ini, /etc/php5/apache2/conf.d/20-mysqli.ini, /etc/php5/apache2/conf.d/20-pdo_mysql.ini, /etc/php5/apache2/conf.d/20-readline.ini
PHP API	20121113
PHP Extension	20121212
Zend Extension	220121212
Zend Extension Build	API220121212,NTS
PHP Extension Build	API20121212,NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend	enabled

WordPressの開発環境

Local by Flywheel

<https://local.getflywheel.com/>

- すぐにローカルでWordPressの環境を構築
- SSHでの接続やMailCatcher、WP-CLIが使える
- 個人的にWordPressの開発環境はこれ一択



Local Sites

- SAMPLE01
- SAMPLE02
- TESTSITE01**

testsite01

STOP SITE

~/Local Sites/testsite01

SITE SETUP DATABASE SSL UTILITIES

ADMIN

VIEW SITE

WordPress Version 4.8.3

PHP Version 7.0.3 RESET CONFIG

Web Server nginx

MySQL Version 5.6.34

Site Name testsite01

Site Domain testsite01.dev [CHANGE](#)

Site Path ~/Local Sites/testsite01

Multisite No



3 sites running

STOP ALL

Live Link: [ENABLE](#)

画像の書き出し

Photoshopのアセット書き出し

- defaultレイヤーの活用
- 最小の記述? : default 100% + 200% @2x

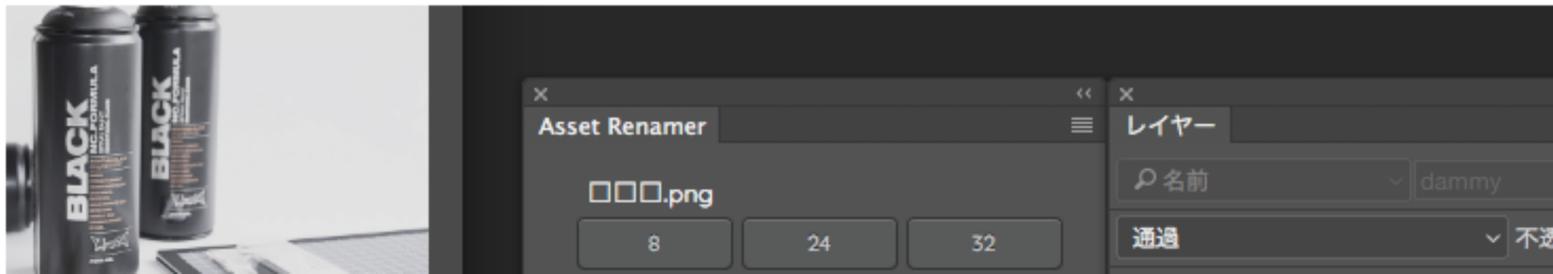
http://dearps.lovwar.com/2017/11/03/asset-renamer/



© 2017.11.03

画像アセット用リネーム エクステンション「Asset Renamer」 CC 2018対応

いいね! 15 ツイート G+ B! 1 Pocket 6 LINEで送る



最近の投稿

画像アセット用リネーム エクステンション「Asset Renamer」 CC 2018対応

Adobe Photoshop CC 2018がリリースされました。

選択範囲をシェイプに変換するスクリプト

DIFF 差分確認レイヤー作成スクリプト

Photoshop CC 2017とAdobe Stockの連携で作業がさらにスピードアップ!

lock

検索

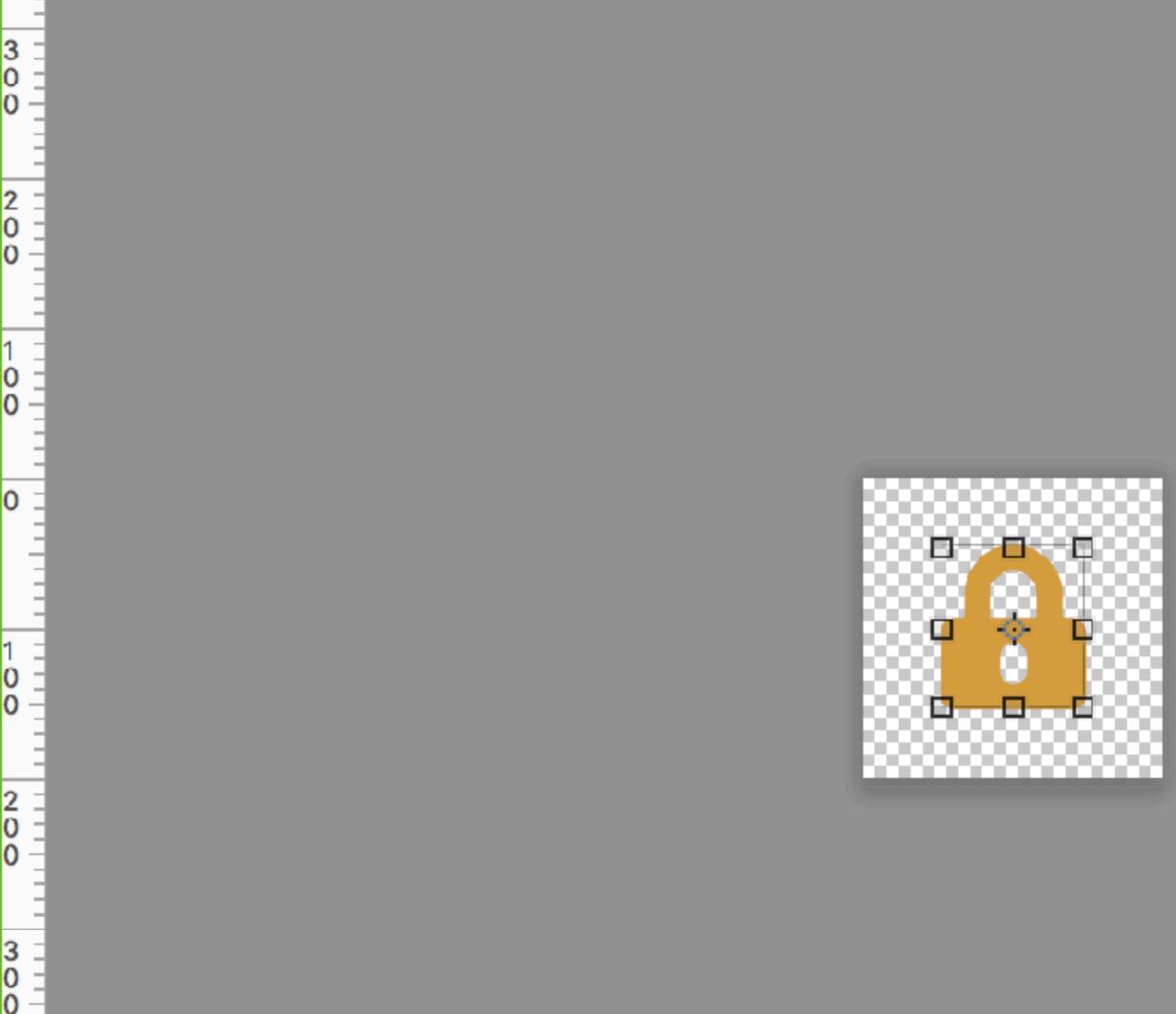
名前	変更日	サイズ	種類
lock-assets	今日 8:37	--	フォルダ
lock.png	今日 8:37	2 KB	PNGイメージ
lock@2x.png	今日 8:37	5 KB	PNGイメージ
lock.psd	今日 8:35	50 KB	Adobe...hop file

Adobe Photoshop CC 2018

自動選択: レイヤー | バウンディングボックスを表示

3D モード: [Icons]

lock.psd @ 100% (lock.png32, RGB/8#) *



Asset Renamer

□□□.png

8 24 32

□□□.png,200% □□□@2x.png

8 24 32

50% □□□.png,□□□@2x.png

8 24 32

□□□.jpg

50 80 100

□□□.jpg,200% □□□@2x.jpg

50 80 100

50% □□□.jpg,□□□@2x.jpg

50 80 100

チャンネル | パス | レイヤー

種類

通常 不透明度: 100%

ロック: [Icons] 塗り: 100%

default 100% + 200% @2x

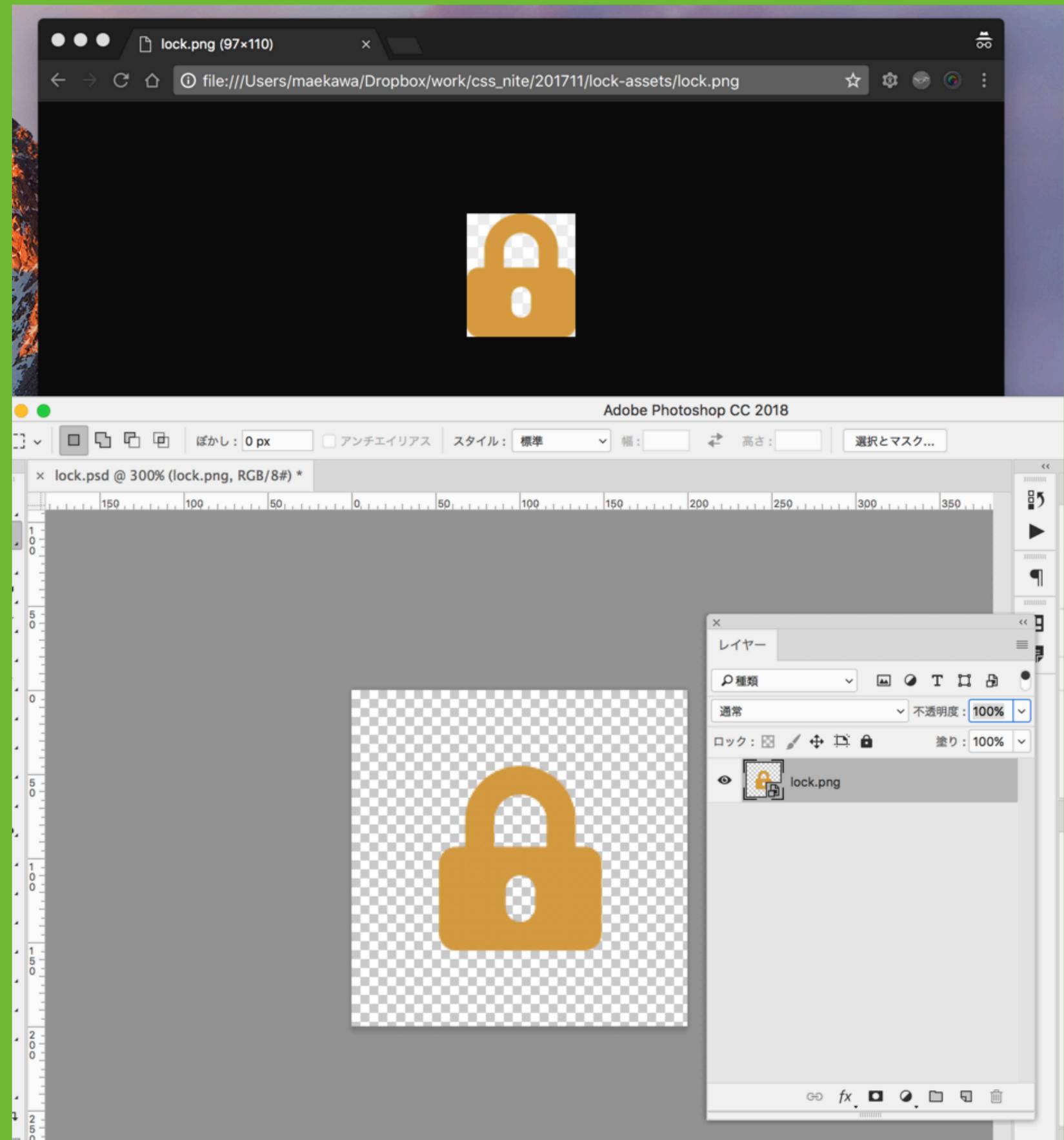
lock.png32

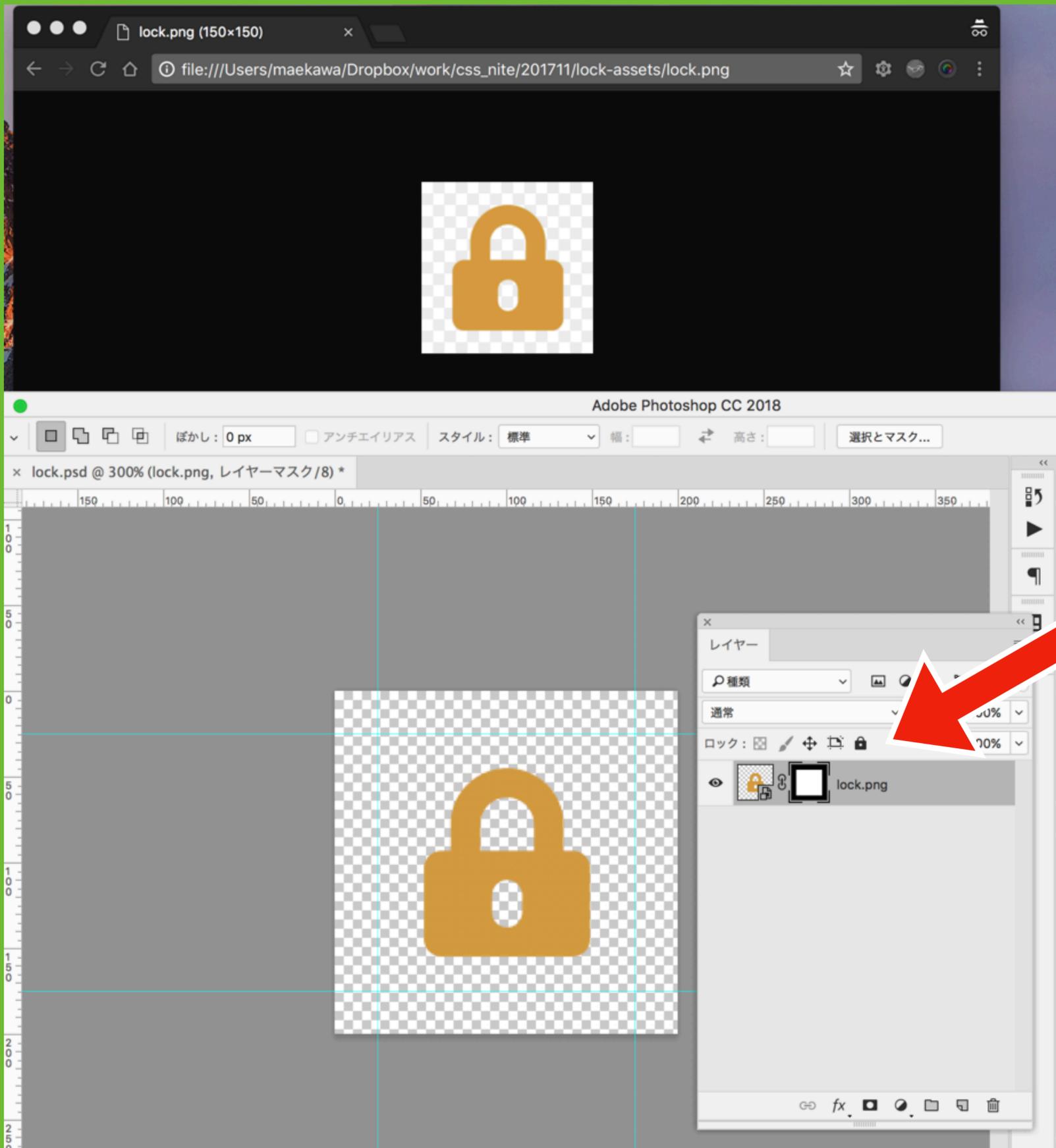
Re

余白が必要な場合

- レイヤーマスクを利用

通常のアセット書き出しでは、
余白が出ないように出力





余白を入れたい場合は
レイヤーマスクを広く入れると、
その範囲で書き出してくれます。

Photoshop入稿で
悲しくなるとき……

乗算

http://graphicartsunit.tumblr.com/post/

75913670229/%E7%99%BD%E3%81%AE%E8%A6%81%E7%B4%A0%E3%82%92%E9%80%8F%E6%98%8E%E3%81%AB%E3%81%99%E3%82%8Bphotoshop%E3%82%A2%E3%82%AF%E3%82%B7%E3%83%A7%E3%83%B3

graphicartsunitをフォローする **tumblr.**

saucer ABOUT ARCHIVE

2月 08, 2014

白の要素を透明にするPhotoshopアクション

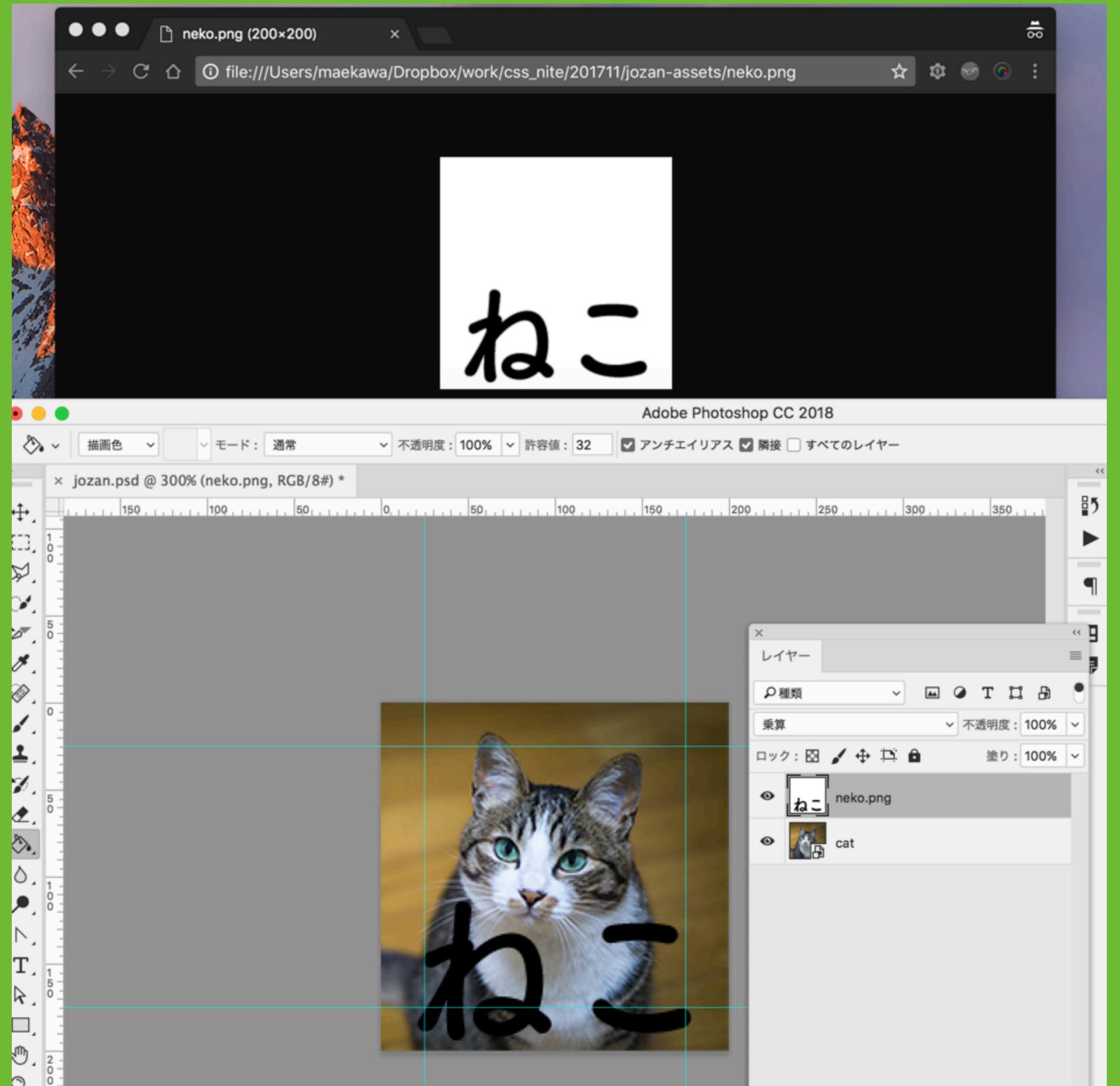
少し前にTwitterでユミソラさん (@yumisora) とロゴの抽出についてお話していて、以前「Photoshop 楽・速・美」という書籍で書いた「炎を奇麗に抽出する」というネタを思い出したので、画像の白い部分を透明にするように内容を展開してアクションにしてみました。興味があれば使ってみてください。(2014.2.8 15:55更新)

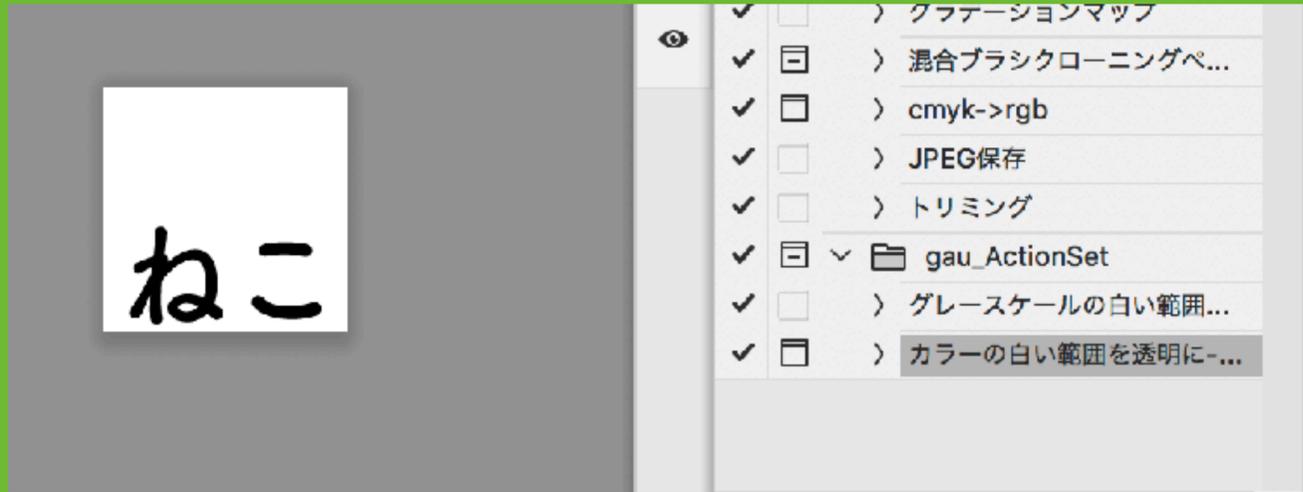
どういう処理ができるか

画像の中の白い部分を透明にします。グラデーションなども半透明でキレイに抜けます。グレースケール画像に限っては、エッジのフリンジ（ギザギザ）もない状態でキレイに抽出が可能です。白黒のロゴや線画をすばやく透明化したときは、便利に使えるんじゃないかと思います。

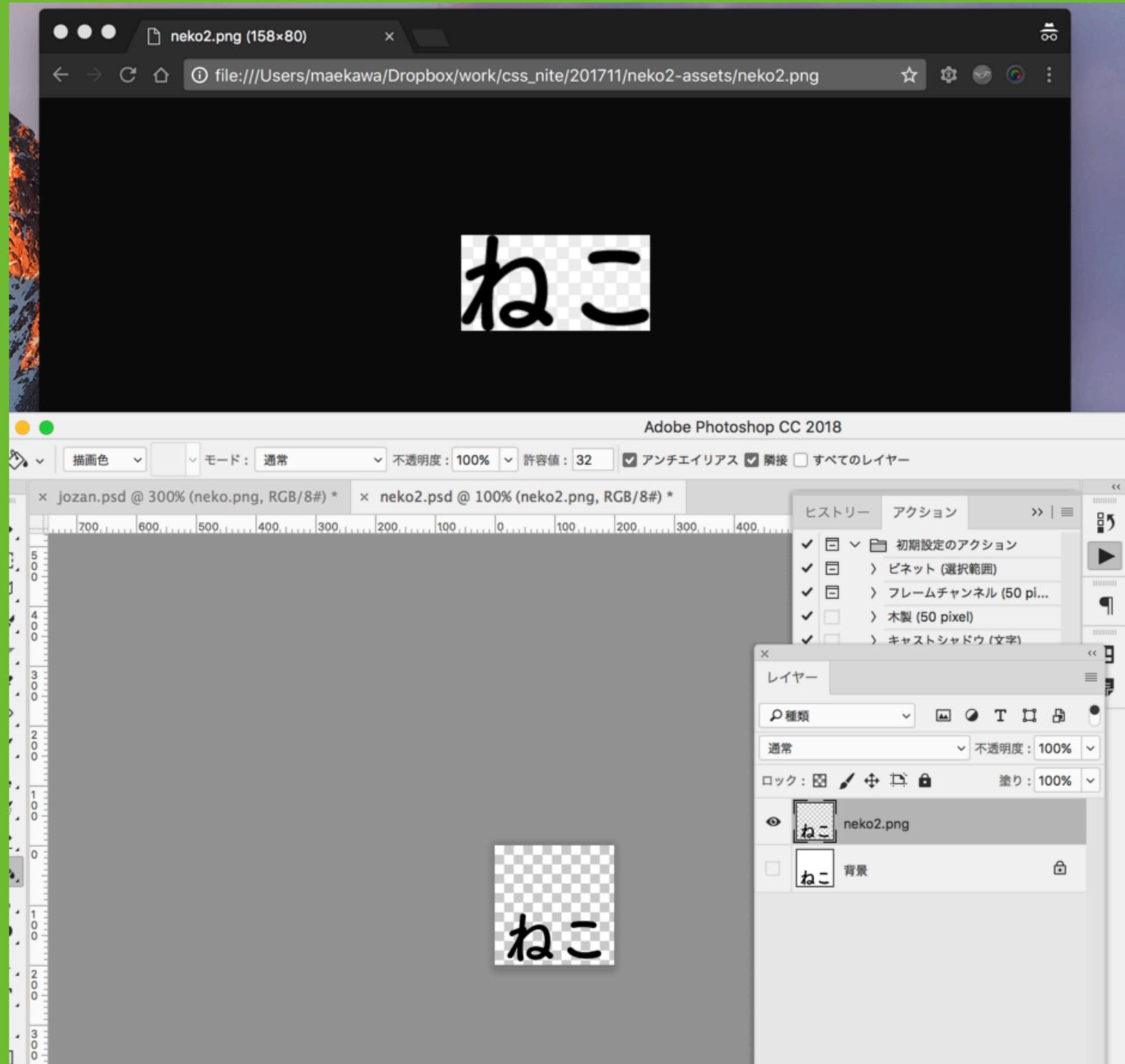
TRANSPARENT → TRANSPARENT

書き出してみても気づく乗算





スクリプトを実行



Illustratorのアセット書き出し

- Illustratorにもアセット書き出しの機能あり
- Photoshopとは異なるUI
- 書き出した画像のデータが比較的重い

書き出した画像が、
ぼやけたり、線が切れたりする場合

書き出しの設定を確認

「スクリーン用に書き出しダイアログを開く」
ボタンを押す



形式の設定

PNG

PNG 8

JPG 100

JPG 80

JPG 50

JPG 20

SVG

PDF

オプション

アンチエイリアス: 文字に最適 (ヒント) ⓘ

なし

アートに最適 (スーパーサンプリング)

背景色:

✓ 文字に最適 (ヒント)

キャンセル

設定を保存

選択:

すべてのアセット

書き出し先:

/Users/...ekawa/work/temp/

書き出し後に場所を開く

サブフォルダーを作成

フォーマット:

iOS Android ⓘ

拡大・縮小 サフィックス 形式

1x

なし

JPG 80

+ スケールを追加

選択を解除

プレフィックス:

選択済み: 1、書き出しの合計: 1

キャンセル

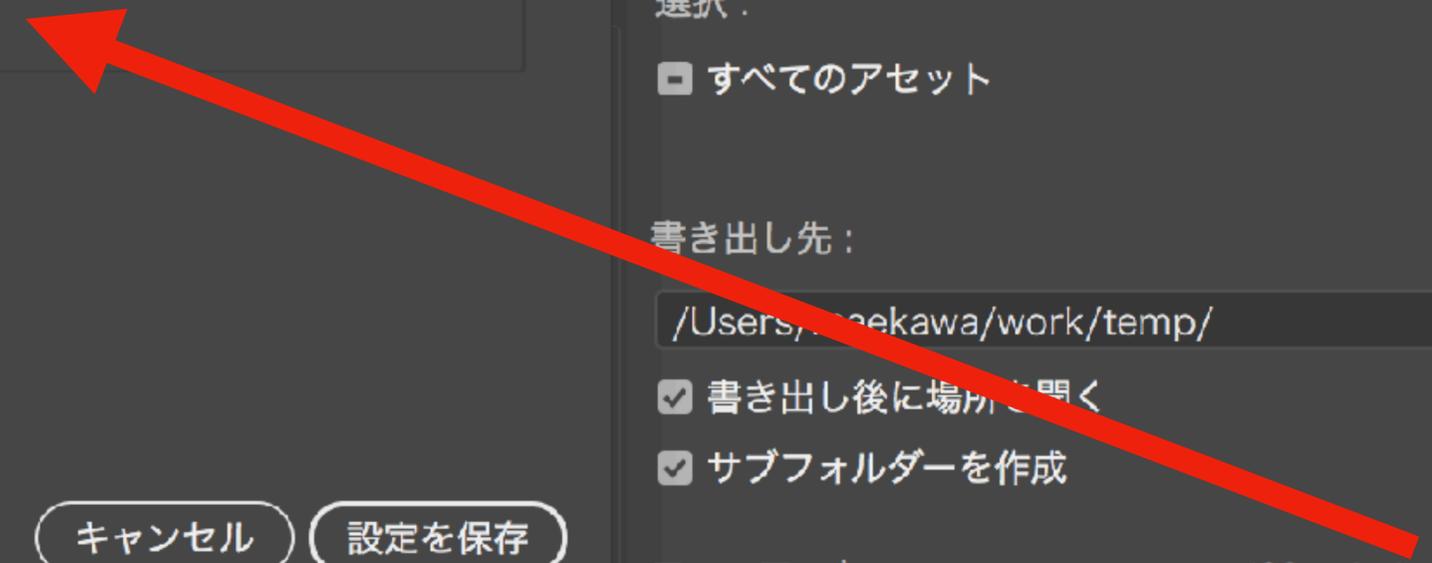
アセットを書き出し



アートワーク 7

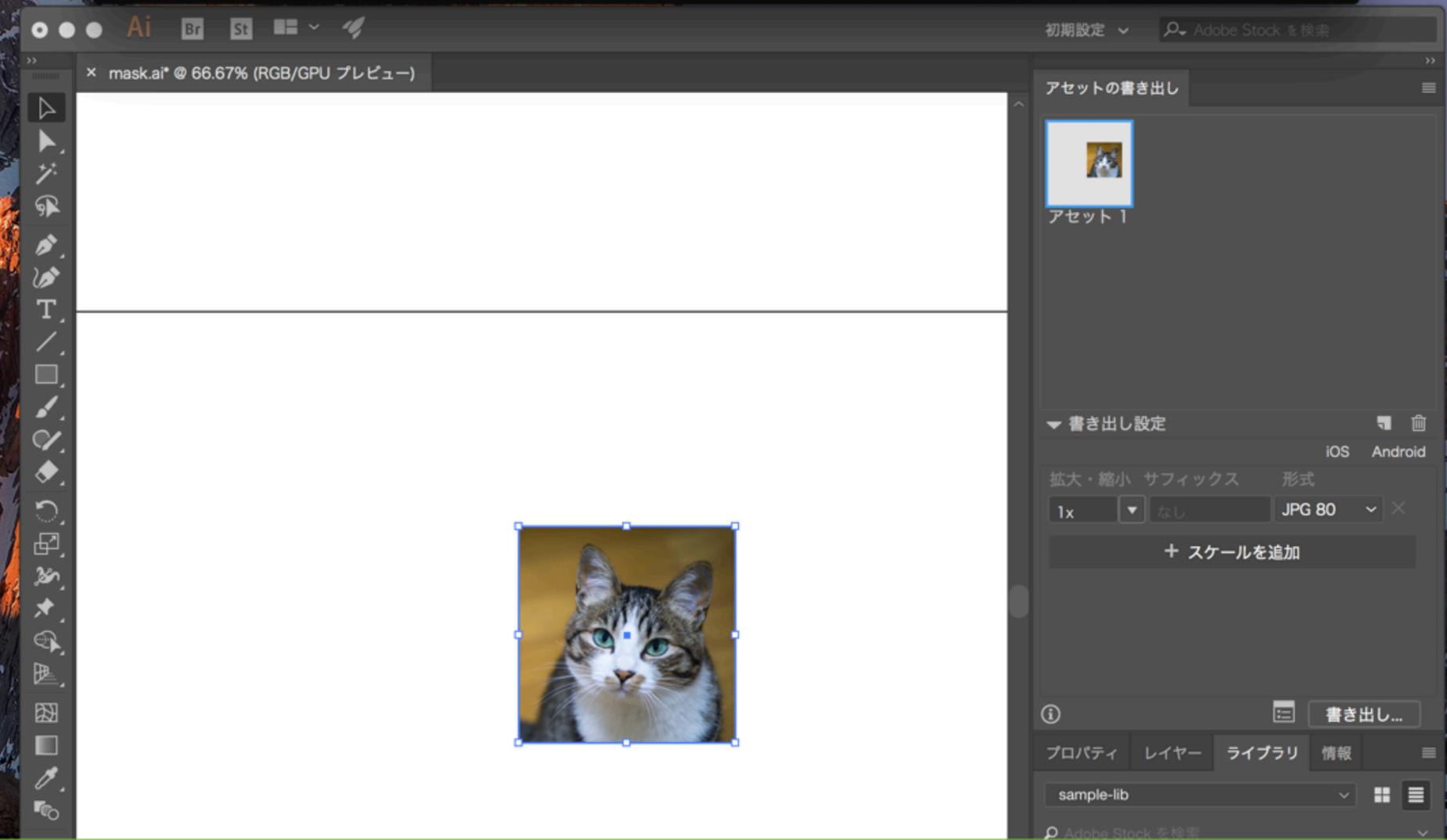
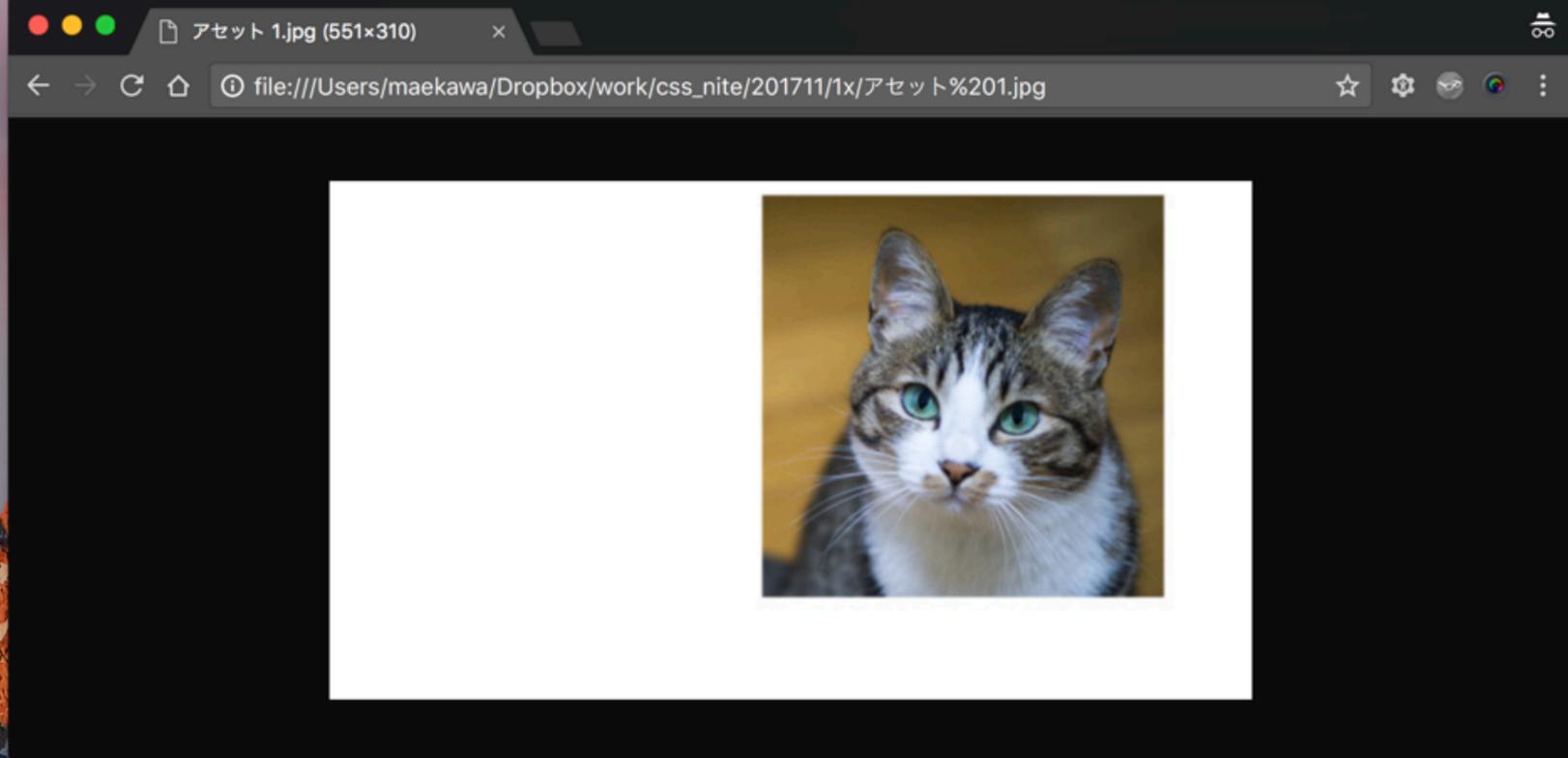


プロパティ レイヤー



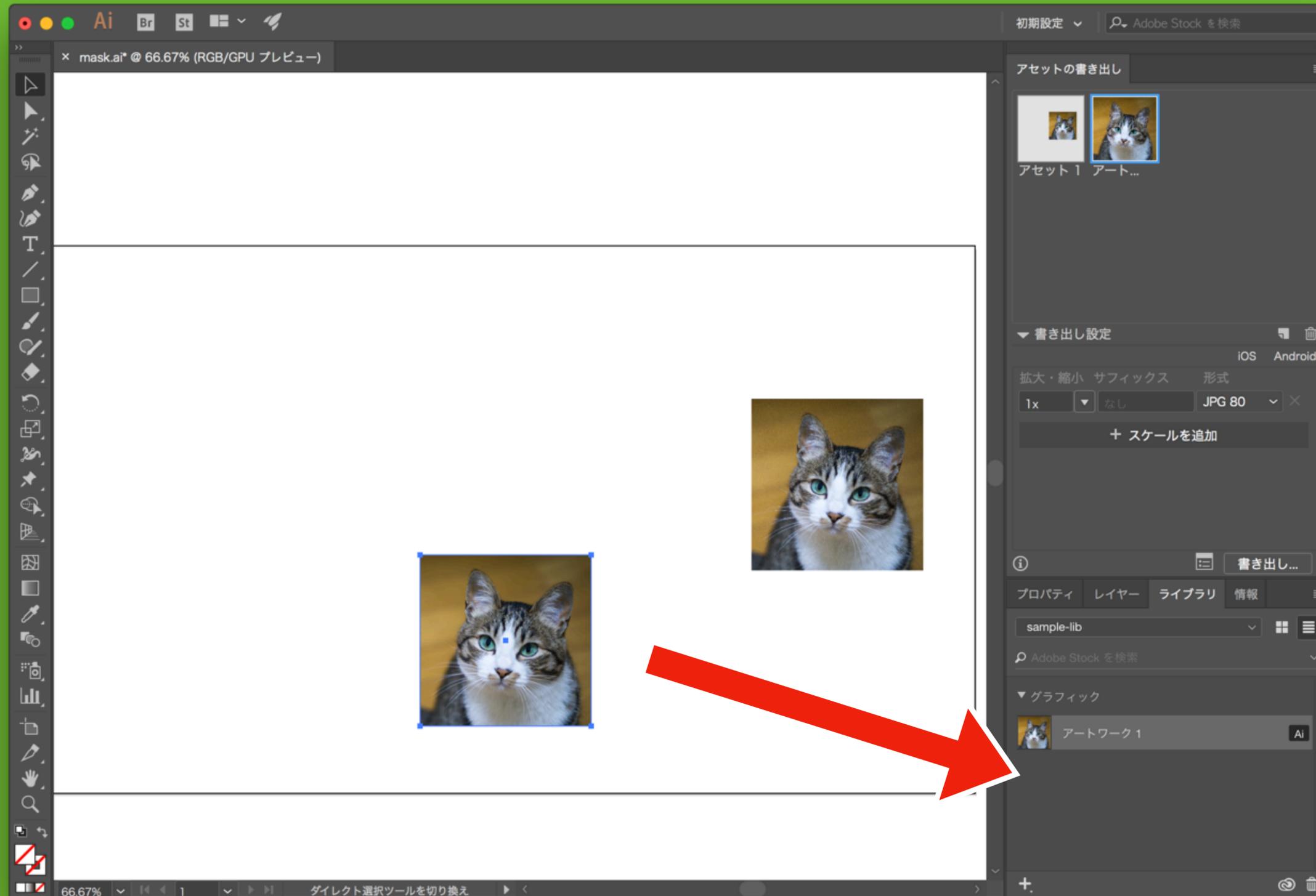
マスクしていても余白が出る

- Illustratorではマスクされている範囲も書き出される
- 一度ライブラリに登録し、それを「書き出し用に追加」
- 表示されている領域で書き出されるように

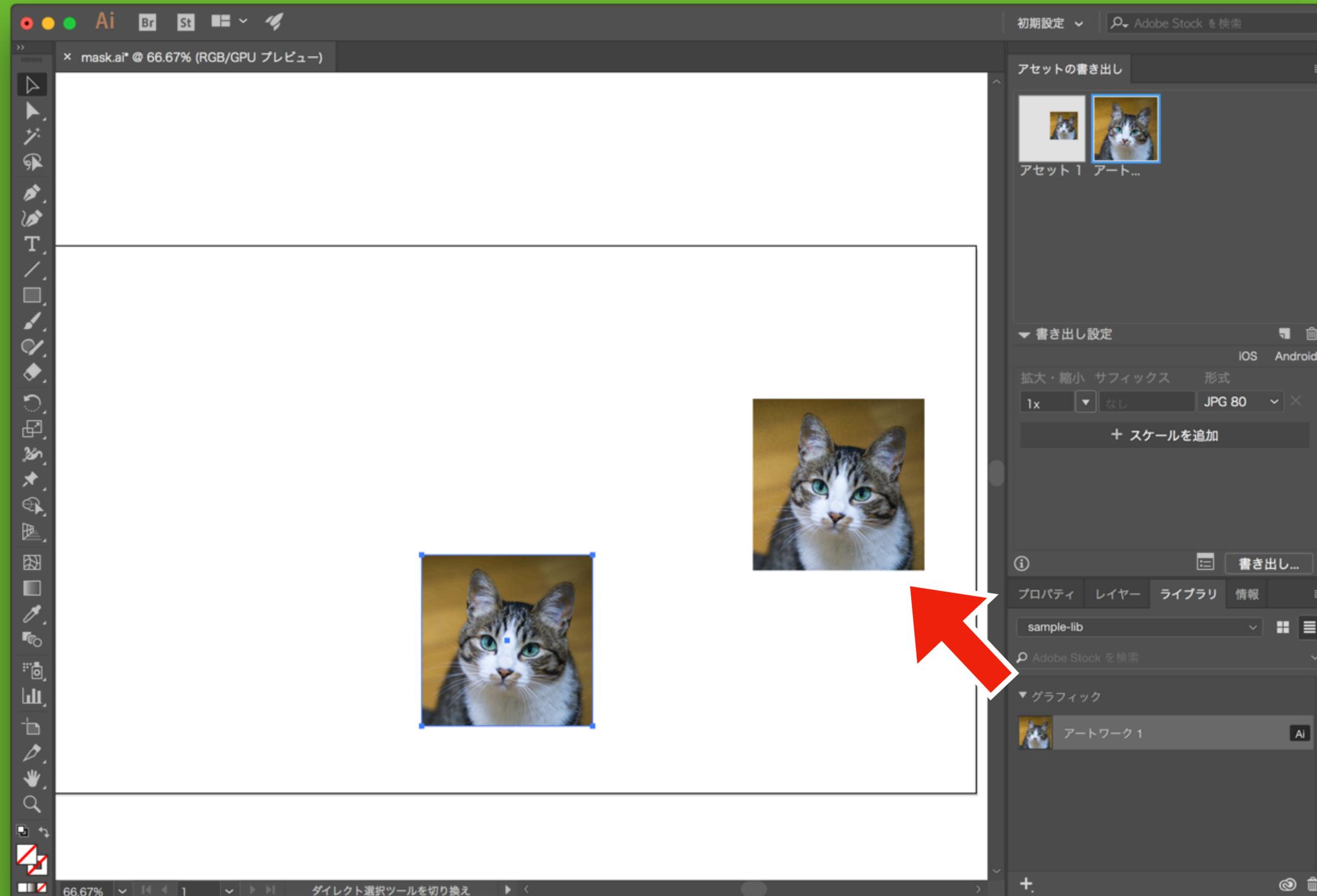


マスクされた写真を
そのままアセットに書き出すと
元の画像のサイズで
余白が出てしまいます

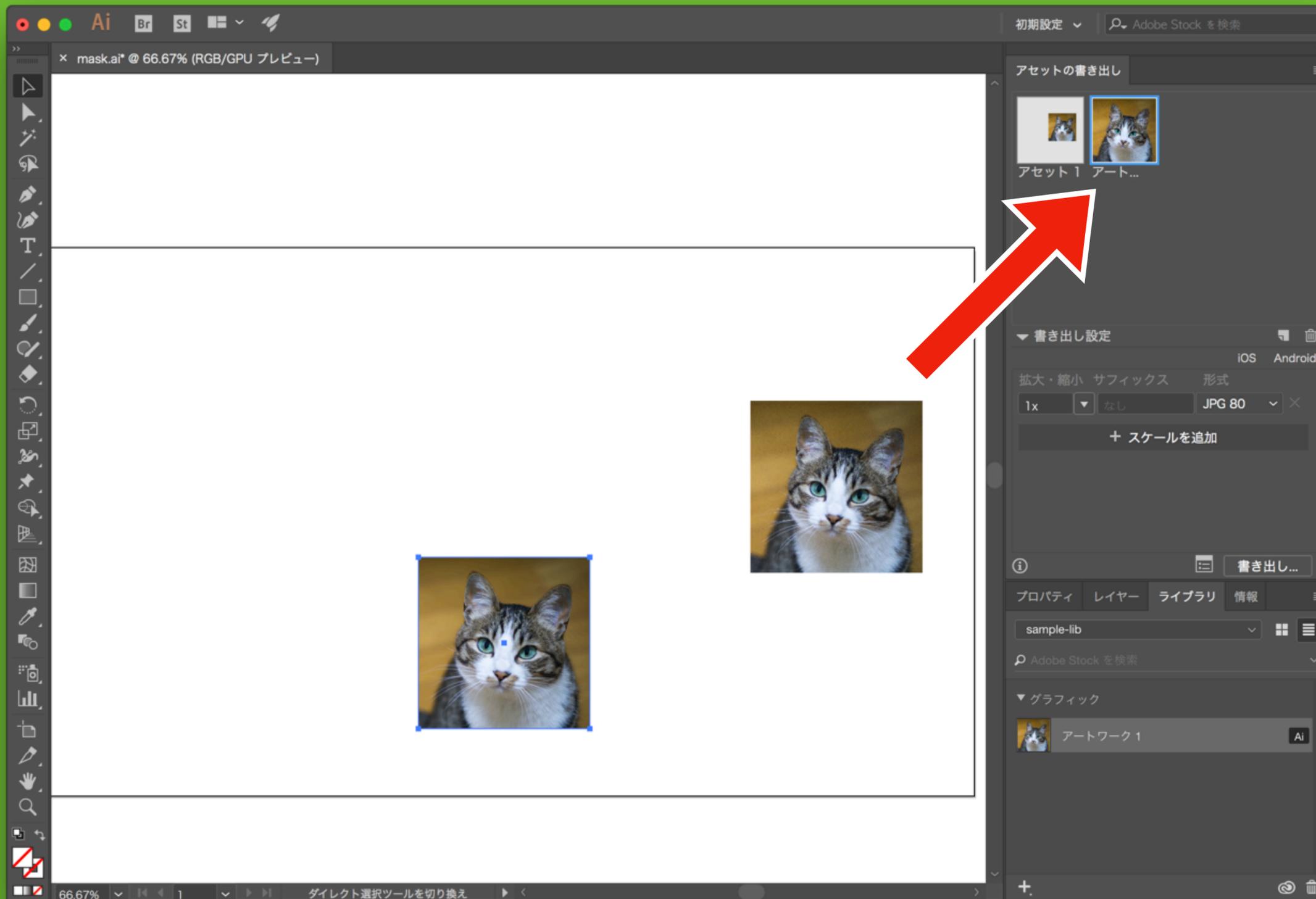
ライブラリに一度登録



ライブラリから配置



アセットの書き出しに登録



https://twitter.com/tah_timing/status/864538798859747328

北村たあさんのツイート: "Illust x

Twitter, Inc. [US] https://twitter.com/tah_timing/status/864538798859747328

ホーム モーメント

アカウントをお持ちの場合 ログイン

北村たあ [@tah_timing](#) フォローする

Illustratorのアセット書き出しは「マスクした画像の余白まで書き出す」という変な動きをするけど、ライブラリにマスクごと登録して、それを配置（リンクで）→アセット書き出しすれば余白がなくなるんだよね。

#誰も気づいてなさそうな小技シリーズ



10:51 - 2017年5月16日

35件のリツイート 66件のいいね

35 66

© 2017 Twitter Twitterについて ヘルプセンター 規約 プライバシーポリシー クッキー 広告について

Illustrator入稿で
悲しくなるとき……

小数点のピクセル

端まできれいに書き出されない



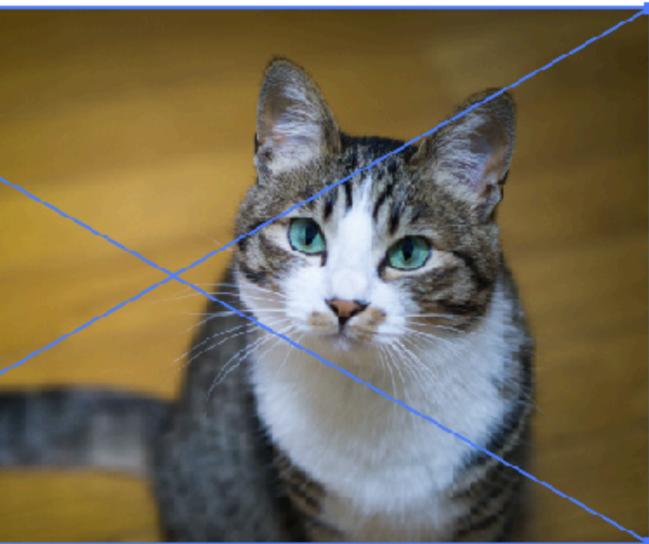
初期設定 ▾ Adobe Stock を検索

プレビュー) × px.ai* @ 100% (RGB/GPU プレビュー)

プロパティ レイヤー ◇ 情報

+ X : 125.65 px W : 303.164 px
Y : 80.62 px H : 170.58 px

アートボード ライブラリ アセットの書き出し



http://3fl.jp/is005

3flab inc.

Notes

Download



Download > Illustrator Scripts > アイテム数値調整ダイアログ

アイテム数値調整ダイアログ

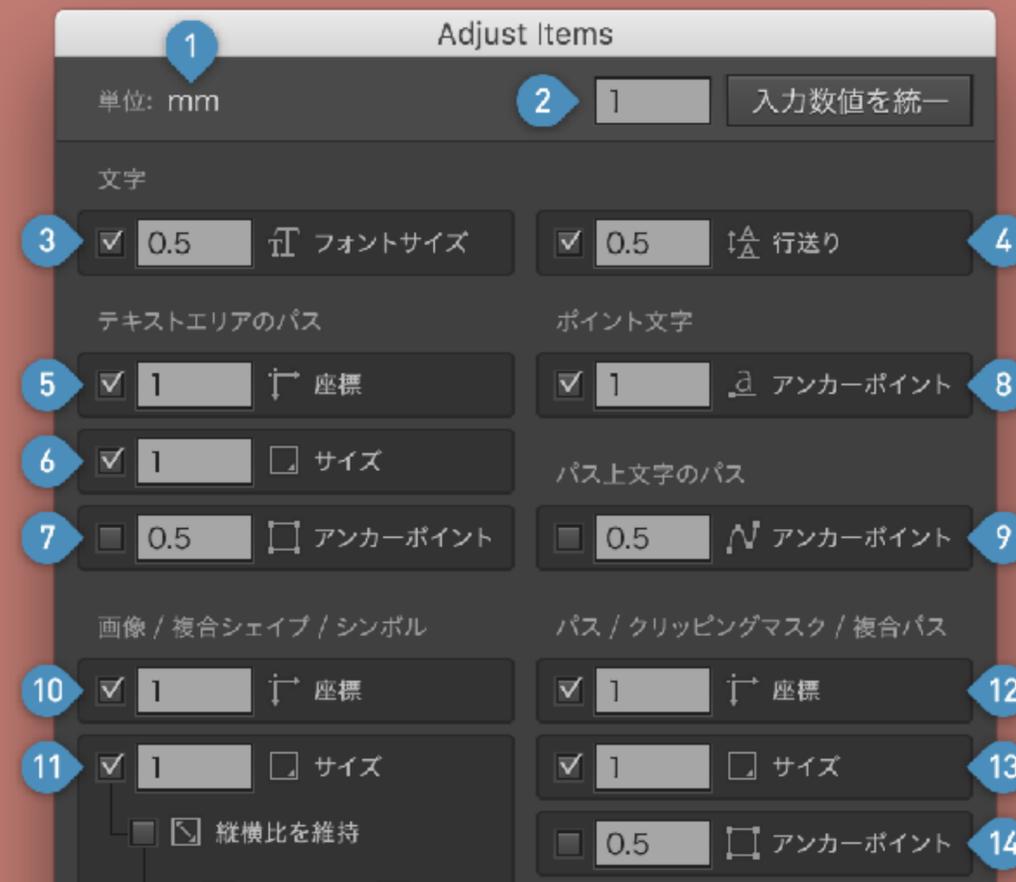
Adjust Items by Dialog

Illustrator Scripts

<http://3fl.jp/is005>

2015-02-02 | 2015-11-27 Update

Adjust Items by Dialog は、選択したオブジェクトの座標やサイズの数値を調整する Illustrator スクリプトです。



 Adjust Items by Dialog 2

調整後



初期設定 ▾ Adobe Stock を検索

プレビュー) × px.ai* @ 100% (RGB/GPU プレビュー)

プロパティ レイヤー ◇ 情報

+ X : 126 px W : 303 px
Y : 81 px H : 171 px

アートボード ライブラリ

1300_トップ ▾

Adobe Stock を検索

- ▶ カラー
- ▶ 文字スタイル
- ▶ グラフィック
- ▶ レイヤースタイル

後作業

デザインとの整合性の確認

PerfectPixel

<https://chrome.google.com/webstore/detail/perfectpixel-by-welldonec/dkaagdgmjmgdmbnecmcefdhjekoceebi>

The image shows a screenshot of the Chrome Web Store page for the 'PerfectPixel by WellDoneCode' extension. The page is in Japanese. At the top, the extension name is displayed with a puzzle piece icon, the provider 'WellDoneCode', and a 'CHROME に追加済み' (Added to Chrome) button. Below this, there are tabs for '概要' (Overview), 'レビュー' (Reviews), 'サポート' (Support), and '関連アイテム' (Related Items). The main content area features a large preview image of the extension's interface, which includes a semi-transparent red overlay with a pixelated pattern and a control panel with sliders for opacity, origin, and scale. To the right of the preview, there is a description in English: 'This extension helps develop your websites with per pixel accuracy! Extension for web developers and markup designers. PerfectPixel allows developers to put a semi-transparent image overlay over the top of the developed HTML and perform pixel perfect comparison between them.' Below the description, there are links for 'お使いの端末に対応' (Compatible with your device), 'ウェブサイト' (Website), and '不正行為を報告' (Report a problem). The '追加情報' (Additional information) section lists the version as 1.88, the update date as October 1, 2017, and the size as 494KiB. At the bottom, there is a carousel of related extensions, including 'Meta Generator Version Check', 'Window Resizer Pro', 'Meta Getty', and 'ショーの画像として Base64'.



ブラウザの表示の上に
画像を透かして表示する機能拡張

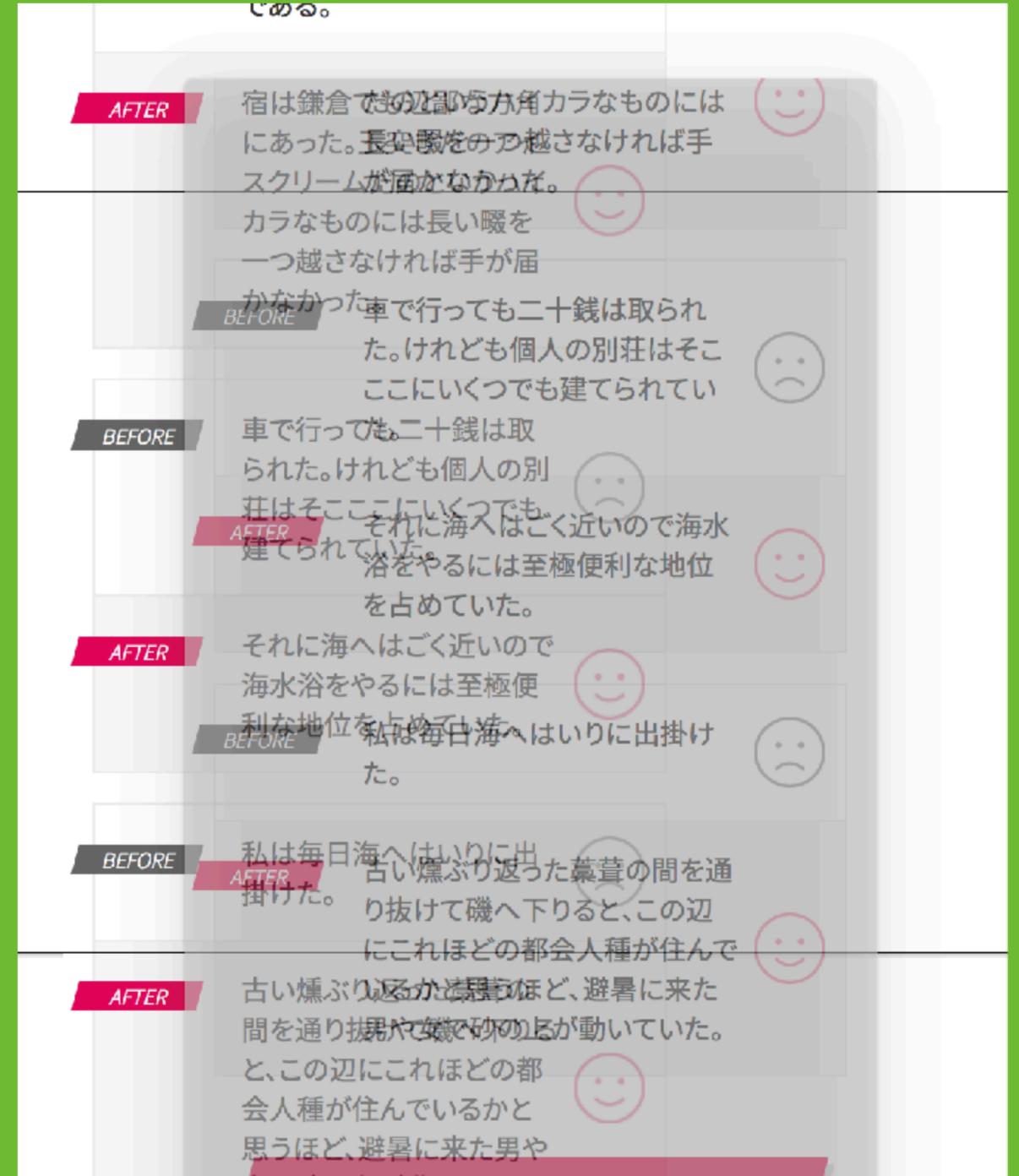
位置の微調整や透明度変更、色の反転
などもできる

Fluid Browser

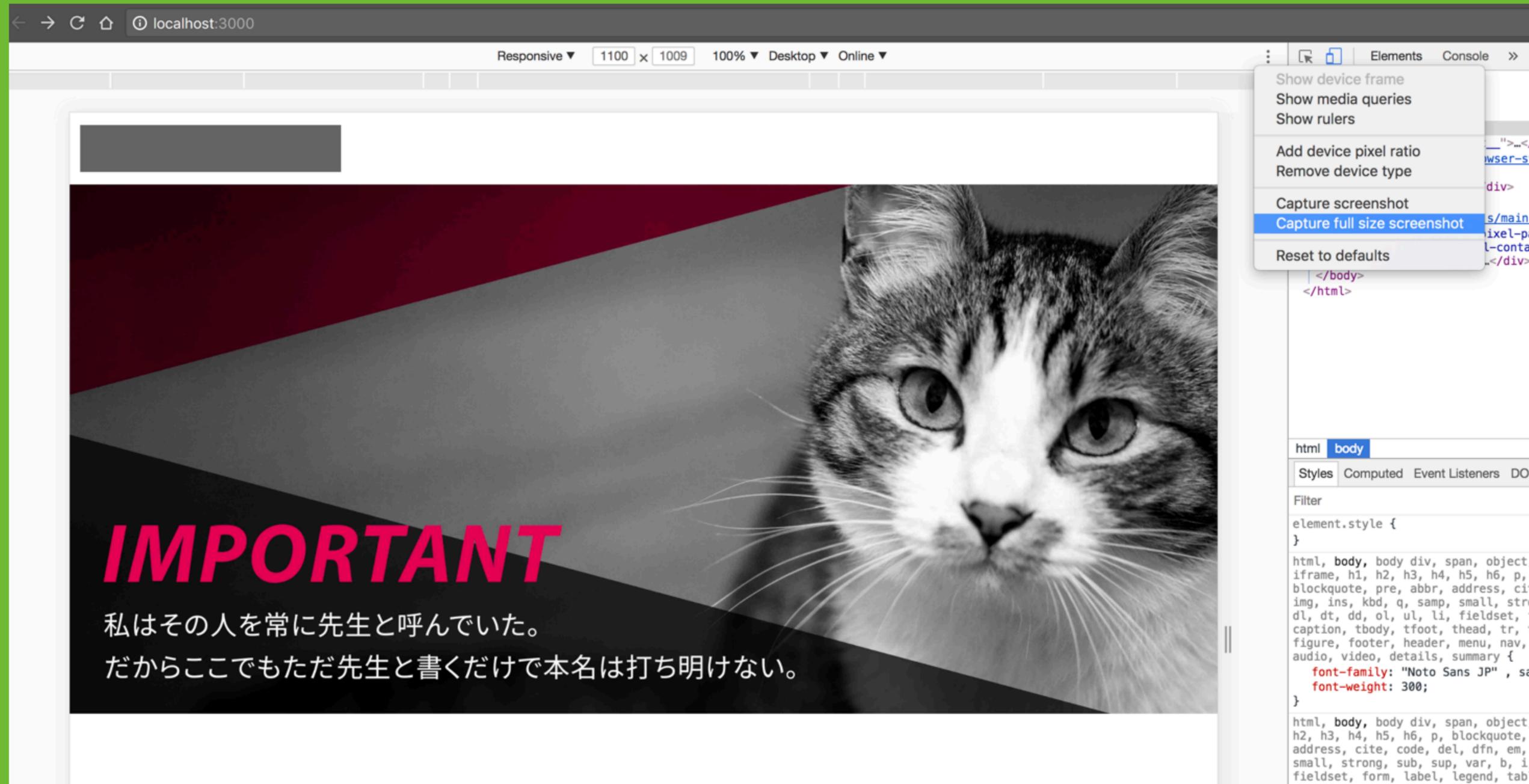
<https://itunes.apple.com/jp/app/fluid-browser/id1077036385>

半透明になるブラウザ

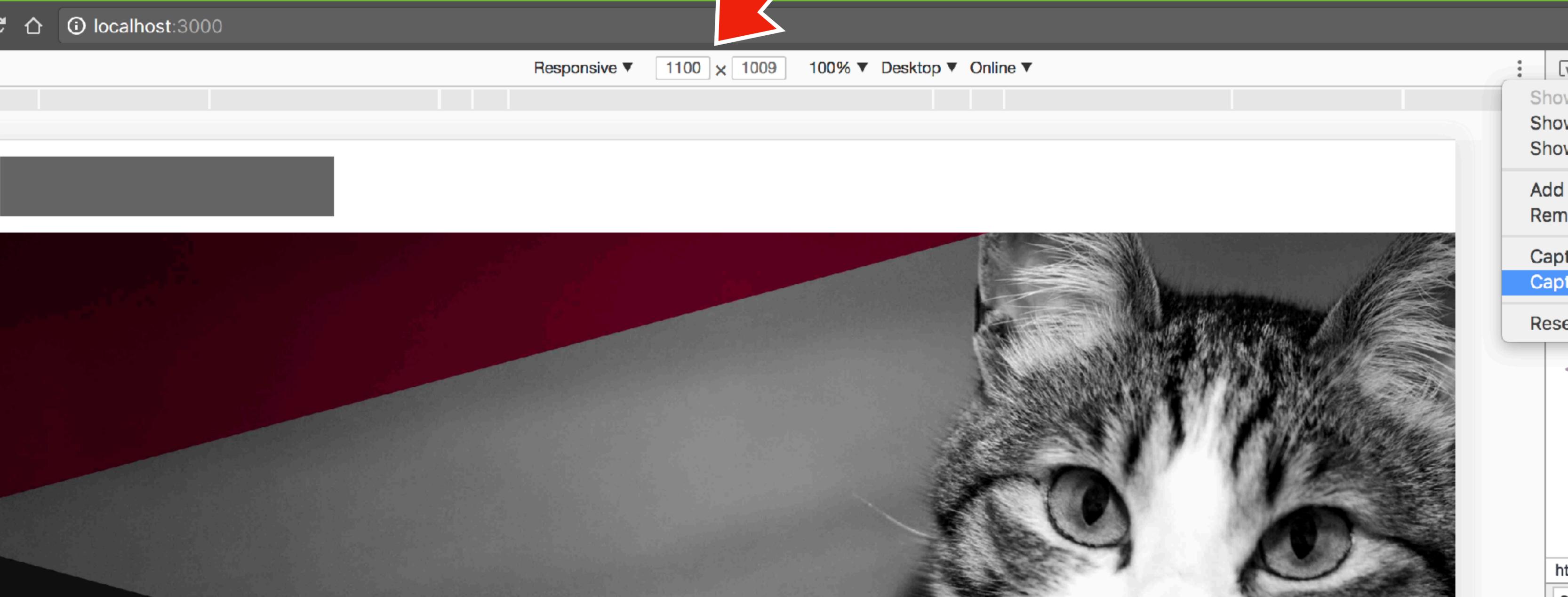
Browsersyncでリアルタイムでの
リロードも可能



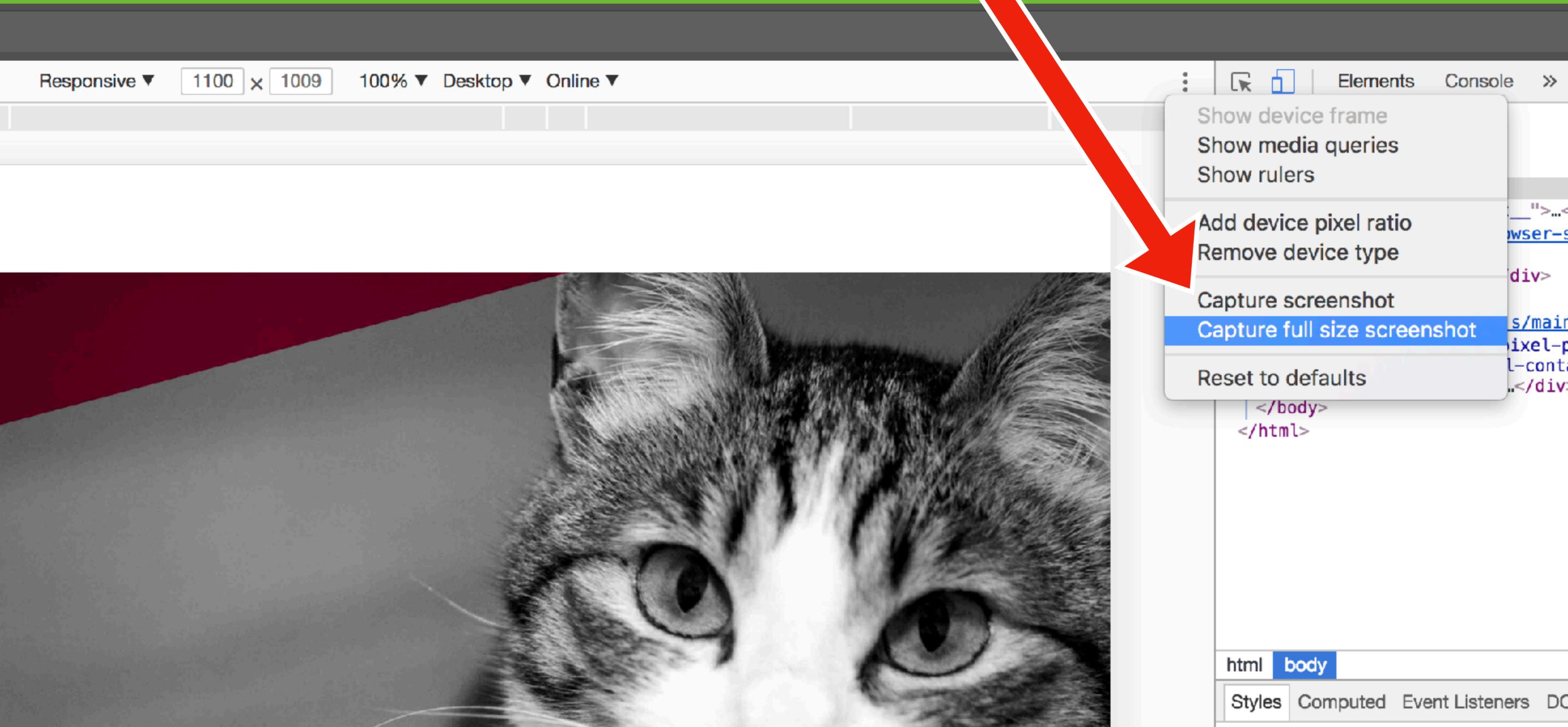
ChromeのDevTools



カンブと幅をあわせて



スクリーンショットを撮る



エミュレーターでの 動作確認

iOSシミュレーター

The image displays two side-by-side screenshots. The left screenshot shows an iOS simulator with a mobile browser displaying a website for 'CSS Nite LP54 Coder's High 2017'. The right screenshot shows a desktop browser (Safari) displaying the source code of the same website, with a file explorer on the left showing the directory structure.

Mobile View (Left):

12:34
cssnite.jp

CSS Nite LP54
Coder's High 2017

LPシリーズの54弾はコーディングをテーマに開催します (企画背景)。

- 11月3日[金]は祝日。開催日の11月4日[土]は、三連休の中日のようです。
- 翌日 (11月5日[日]) に**Gitのガッツリ講座を連動企画**いただきました。
- 事前決済での受付を終了しました。これ以降、**当日枠のみ受け付け**ます。

★ 概要

イベント名
CSS Nite LP54 「Coder's High 2017」

日時
2017年11月4日 (土) 13:20-19:45 (開場)

Desktop View (Right):

Webインスペクタ — Simulator — Safari — cssnite.jp — lp54

すべてのリソース 書類

lp54

```
1 <!doctype html>
2 <html lang="ja">
3 <head>
4 <meta charset="UTF-8">
5 <title>CSS Nite LP54 「Coder's High 2017」 (2017年11月4日開催) </title>
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <link rel="stylesheet" href="http://cssnite.jp/css/lp2013.css">
8 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.4.0/css/font-awesome.min.css">
9 <!-- OGP -->
10 <meta property="og:title" content="CSS Nite LP54 「Coder's High 2017」 (2017年11月4日開催)">
11 <meta property="og:type" content="article">
12 <meta property="og:url" content="http://cssnite.jp/lp/lp54/">
13 <meta property="og:image" content="http://cssnite.jp/images/OGP-CSSNiteLP54.png">
14 <meta property="og:image" content="http://cssnite.jp/images/banner-CSSNiteLP54.png">
15 <meta property="og:site_name" content="CSS Nite公式サイト">
16 <meta property="og:description" content="「Coder's High 2017」をテーマに、2017年11月4日 (土) にベルサール半蔵門で開
17 <meta property="fb:app_id" content="169750309753490">
18 <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.7.1/jquery.min.js"></script>
19 <script src="http://cssnite.jp/js/page-scroller.js"></script>
20 <script src="http://cssnite.jp/js/winks.js"></script>
21 <link rel="stylesheet" href="../css/slicknav.css" />
22 <script src="../js/jquery.slicknav.min.js"></script>
23 <script src="http://cdnjs.cloudflare.com/ajax/libs/modernizr/2.6.2/modernizr.min.js"></script>
24 </head>
25 <body class="lp54 blue">
26 <div id="main">
27 <h1>LPシリーズの54弾はコーディングをテーマに開催します ( <a href="#background">企画背景</a> ) 。 </p>
29 <ul>
30 <li>11月3日[金]は祝日。開催日の11月4日[土]は、三連休の中日のようです。 </li>
31 <li>翌日 (11月5日[日]) に<strong><a href="https://coedo-dev.doorkeeper.jp/events/65616">Gitのガッツリ講
32 <li>事前決済での受付を終了しました。これ以降、 <a href="#apply">当日枠のみ受け付け</a>ます。 <br>
33 </li>
34 </ul>
35 <!-- <h3>キャンセル規定について</h3>
36 <p>FAQに記載していますが改めて、ほぼノーリスクなので「予定が見えない」方も早めにお申し込みがおススメです。 </p>
37 <ul>
38 <li>セミナー開催の3日前まで「ビデオ参加費を除いて返金」としてありますが、「ビデオ参加は不要」の方には、全額返金しています。 <
39 <li>懇親会費については、セミナー開催の3日前まで全額返金します。 </li>
40 </ul> -->
41 <section id="outline">
42 <h2><i class="fa fa-star fa-fw"></i>概要</h2>
43 <table class="event-info">
44 <tr>
45 <th>イベント名</th>
46 <td>CSS Nite LP54 「Coder's High 2017」 </td>
47 </tr>
48 </table>
```

Genymotion

<https://www.genymotion.com/>

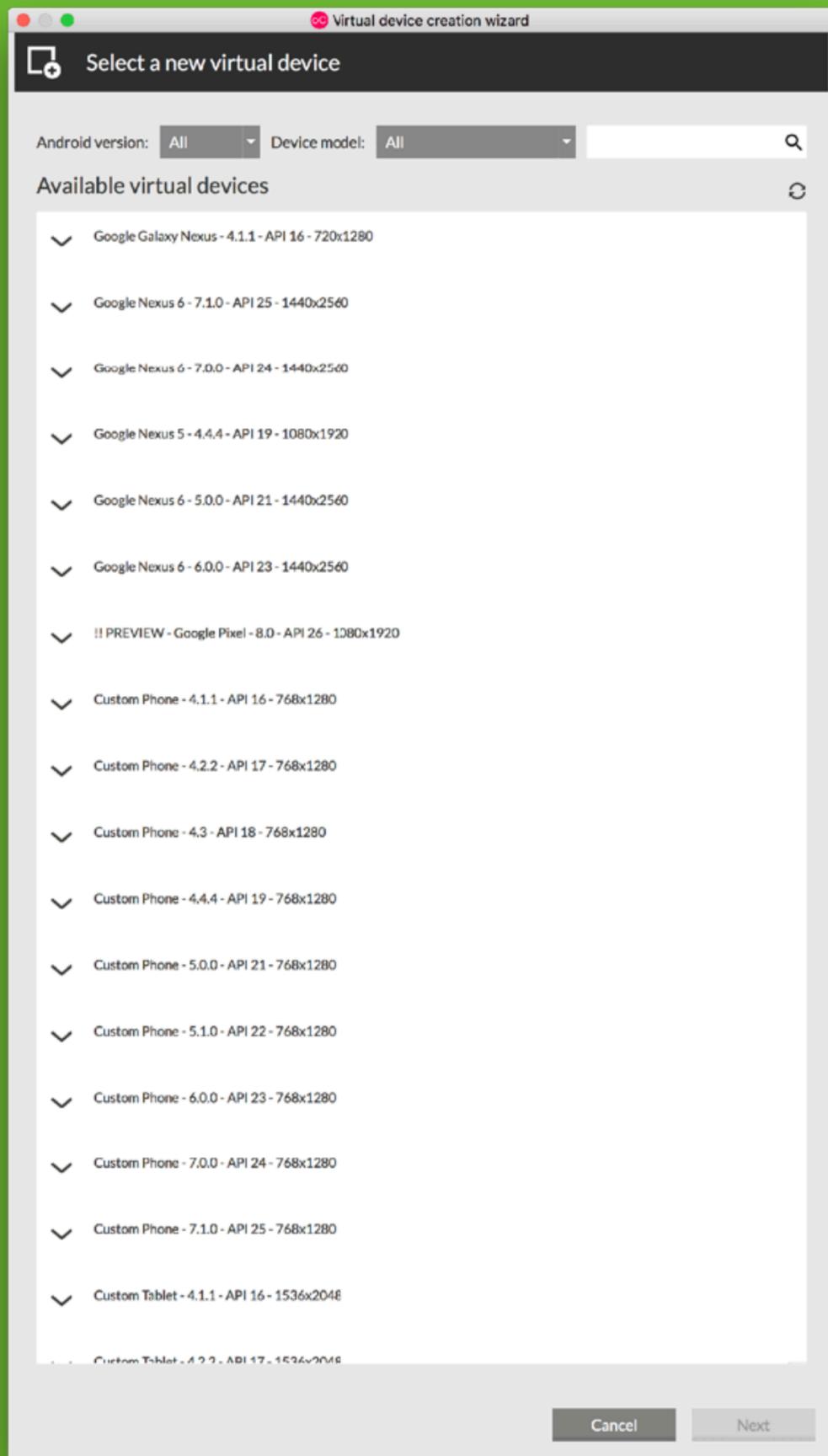
GENYMOTION ^{OO} by 

[Solutions](#) [Pricing](#) [Help](#) [Resources](#) [Other Products](#) [Contact Us](#) [Trial](#) [Sign In](#)

More than an Android emulator, a powerful virtualization platform to address all your professional needs

Develop – Test – Deliver

[Watch the Video](#) [Trial](#) [+](#)



様々なバージョンのAndroidの
エミュレーターを起動できる

ChromeのDevToolsでのデバッグもできる

The image displays a mobile browser interface on the left and the Chrome DevTools interface on the right, both showing the same web page: <http://cssnite.jp/lp/lp54/>. The page content includes the CSS Nite LP54 logo, the title "Coder's High 2017", and a list of event details.

Mobile Browser View (Left):

- WebView Browser Tester 52.0.2743.100
- Address bar: <http://cssnite.jp/lp/lp54/>
- Page content: "Coder's High 2017" header, event description, and a list of dates and activities.
- Bottom navigation bar with icons for home, back, forward, and search.

Chrome DevTools View (Right):

- Developer Tools - cssnite.jp/lp/lp54/
- Elements panel: Shows the HTML structure of the page, with the `<h1>` element selected. The DOM tree shows the following structure:

```
<!DOCTYPE html>
<html lang="ja" class=" js flexbox canvas canvastext webgl touch geolocation postmessage websqlindex indexeddb hashchange history draganddrop websockets rgba hsla multiplebgs backgroundsize borderimage borderradius boxshadow textshadow opacity cssanimations csscolumns cssgradients cssreflections csstransforms csstransforms3d csstransitions fontface generatedcontent video audio localstorage sessionstorage webworkers applicationcache svg inlinesvg smil svgclippaths">
  <head>...</head>
  <body class="lp54 blue">
    <div class="slicknav_menu">...</div>
    <div id="main">
      <h1>...</h1>
      <p>...</p>
      <ul>...</ul>
    </div>
  </body>
</html>
```
- Styles panel: Shows the computed styles for the selected `h1` element, including margin and padding values for different screen sizes.
- Console panel: Shows the console log with "top" selected and "Preserve log" checked.

Google Analyticsなどの イベントトラッキング

こんなonclickの経験は？

```
<a href="/hoge/" target="_blank" onclick="_gaq.push(['_trackEvent',  
'trackCategory', 'trackAction', 'trackLabel']);ga('send',  
{'hitType':'event', 'eventCategory': 'trackCategory', 'eventAction':  
'trackAction', 'eventLabel': 'trackLabel'});dataLayer.push({'event':  
'Click', 'eventCategory': 'trackCategory', 'eventAction':  
'trackAction', 'eventLabel': 'trackLabel'});">
```

旧GA、
ユニバーサルアナリティクス、
タグマネージャー
が混在

まず整理

```
<a href="/hoge/"  
target="_blank"  
data-category="trackCategory"  
data-action="trackAction"  
data-label="trackLabel">
```

必要なものをJSで取得

```
1 var universalGA, classicGA, googleTagManager
2
3 if (typeof ga === "function")
4     {universalGA = true}
5
6 if (typeof _gaq !== "undefined" && typeof _gaq.push === "function")
7     {classicGA = true;}
8
9 if (typeof dataLayer !== "undefined" && typeof dataLayer.push === "function")
10    {googleTagManager = true;}
```

必要なものだけ送信 (例:jQuery)

```
function sendTracking(){
  var $t = $(this),
  categoryCode = $t.attr('data-category'),
  actionCode = $t.attr('data-action'),
  labelCode = $t.attr('data-label')
  if (universalGA){
    ga('send', {
      'hitType': 'event',
      'eventCategory':
        categoryCode, 'eventAction':
        actionCode, 'eventLabel': labelCode})
  }
  if (classicGA){
    _gaq.push(['_trackEvent', categoryCode, actionCode, labelCode])
  }
  if (googleTagManager){
    dataLayer.push({
      'event': 'Click',
      'eventCategory': categoryCode,
      'eventAction': actionCode,
      'eventLabel': labelCode})
  }
}
```

まとめ

前作業	仕様の確認	しっかりやる！
	作業環境の構築	チームの、未来の自分のために そして効率良く
	画像の書き出し	コーダーにもありがたい機能 たくさんあります

後作業	デザイン反映 チェック	ざっくりでも、きっちりでも 目で確認できるとわかりやすい
	エミュレーター	「実機がない！」でも ある程度までは対応を
	トラッキング コード	コードを見通し良く

コードを書く以外にも

- 様々なTipsやツール、方法があります
- なにかひとつでも「知って得した」がありましたか？
- あなたも他の人にとって「知った得した」があるはず！
- よかったら教えてください

ありがとうございました

CSS Nite LP54「Coder's High 2017」

「コーダーの前仕事、後仕事」前川昌幸