



# 第69回PHP勉強会

## php5-gd で画像を弄る話

2013/06/22-

“よや” <yoya@awm.jp>

# 自己紹介

- <http://d.hatena.ne.jp/yoya/> 
- 某社の画像サーバのメンテナンス要員
  - 画質とパフォーマンスチューニングを少々
- 所属は秘密ですが、↓この ImageMagick を弄ってる人と名前が同じ
  - GIF アニメ生成は本当に GraphicsMagick で行うべきか？
    - <http://labs.gree.jp/blog/2013/05/8132/> 
- 画像の検証や説明資料を作るのに、GD をよく使うのと、PHP らしいという事で GD のお話をします。

# 発表の目的

- php-gd を使った事がない人向けに紹介
- 使った事がある人には厳しいツツコミに期待

注)

- 今回は簡単な事が簡単に出来るという話
- 本格的な画像処理(OpenCV 的なもの)の話はしません。

# もくじ

- GD概要
  - 導入
  - 入出力イメージ
- サンプルコード
  - 画像ファイル入出力
  - 画像をいちから生成
- 応用サンプル
- PHP5.5



# GD について

- <http://www.boutell.com/gd/> (old page)
- <http://libgd.bitbucket.org/> (new page)
- GD は動的画像生成のライブラリ
- gd1.0 では “gif draw” の略
- gd1.0 の後半 Unisys LZW 圧縮特許で GIF をサポート外にしてから “graphics draw”
- ちなみに特許切れにより gd 2.0.28 で再び GIF をサポートした

# gif draw?

- gd1.0 の当初は gif 出力が目的
  - palette 形式を前提にした API
  - 途中で jpeg, png 等にも対応
  - (後付けで) true color 対応、alpha channel 対応。
- 
- といった経緯を知っていると、使ってて色々ピンと来る

# GD の導入

- PHP の gd extension を有効にします

- 確認

```
% php -i | grep GD
GD Support => enabled
GD Version => bundled (2.1.0 compatible)
```

- インストール

- Debian

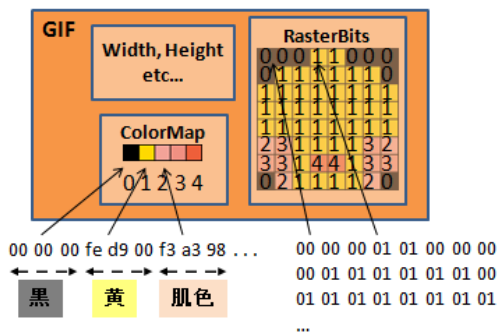
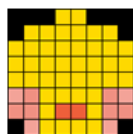
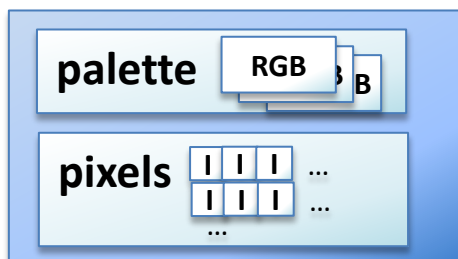
```
% sudo apt-get install php5-gd
```

- 設定ベタ書き (php.ini)

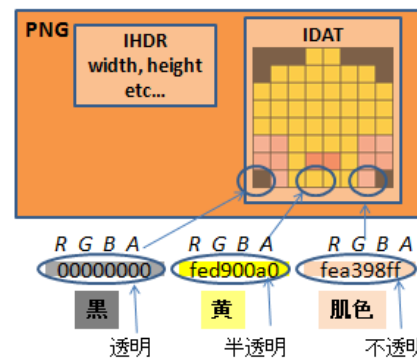
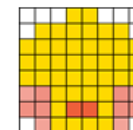
```
;extension=php_gd2.dll
```

# (前提知識) palette と true color

palette  
(pseudo color)



true color  
(direct color)



- 画像引用元) <http://labs.gree.jp/blog/2010/10/1263/> GIF
- <http://labs.gree.jp/blog/2010/12/1902/> PNG

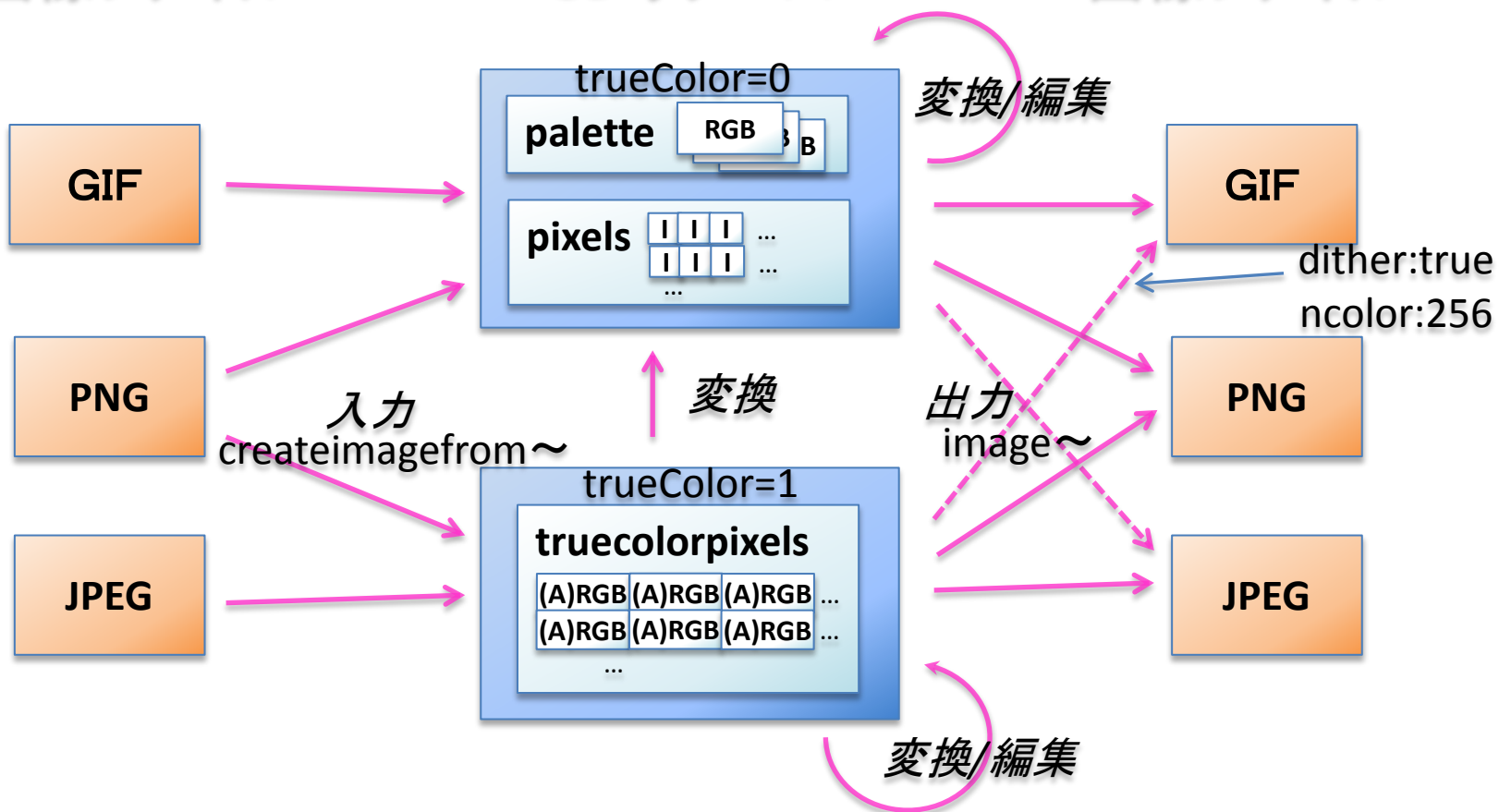


# 入出力フロー

画像ファイル

GD リソース

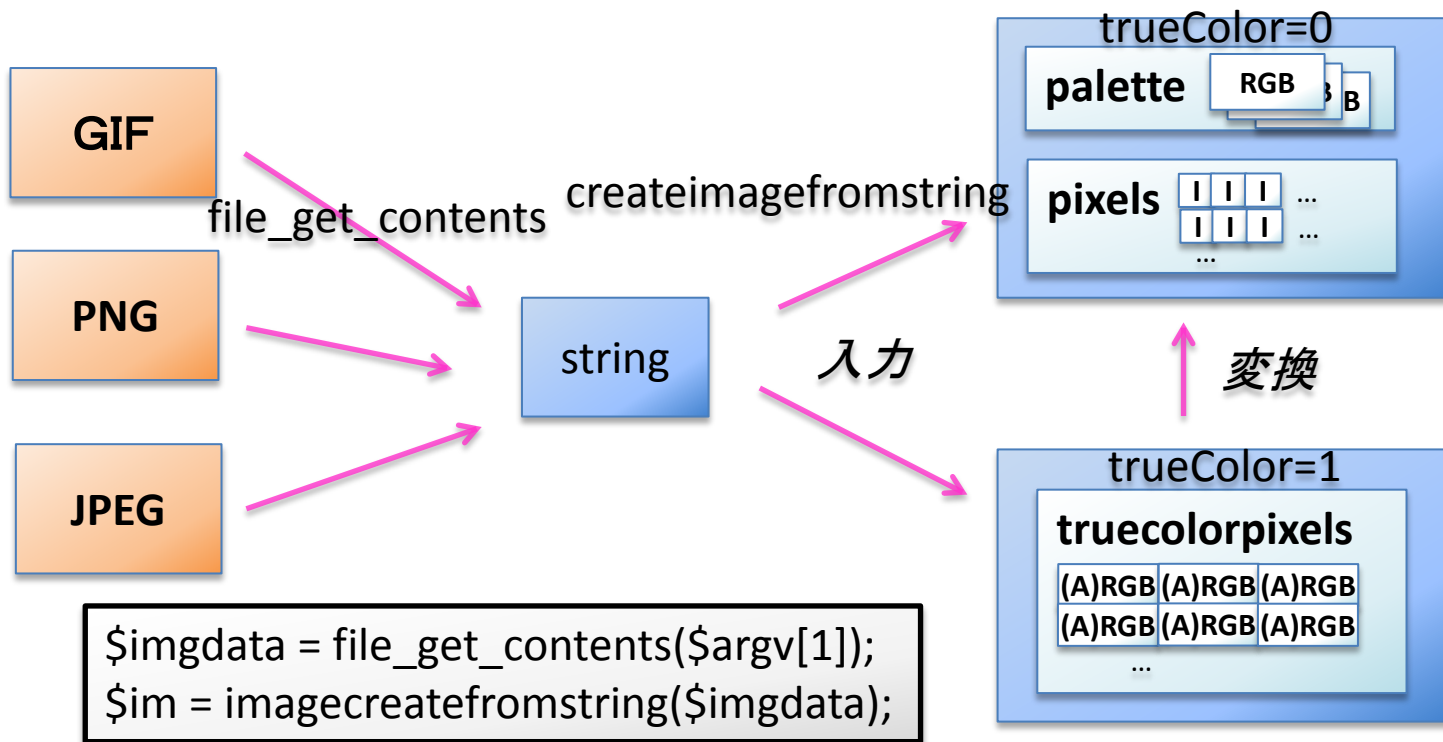
画像ファイル



# ファイル自動判別

画像ファイル

GD リソース



# 画像入出力サンプル

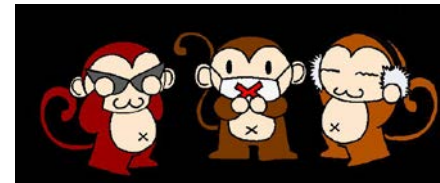
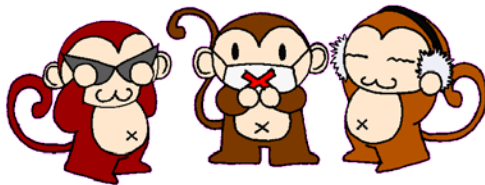
- 入力して出力するだけ



```
<php  
$im = imagecreatefromgif($argv[1]);  
imagejpeg($im);
```

```
% php gif2jpeg.php tomo_pr_263.gif > tomo_pr_263.jpg
```

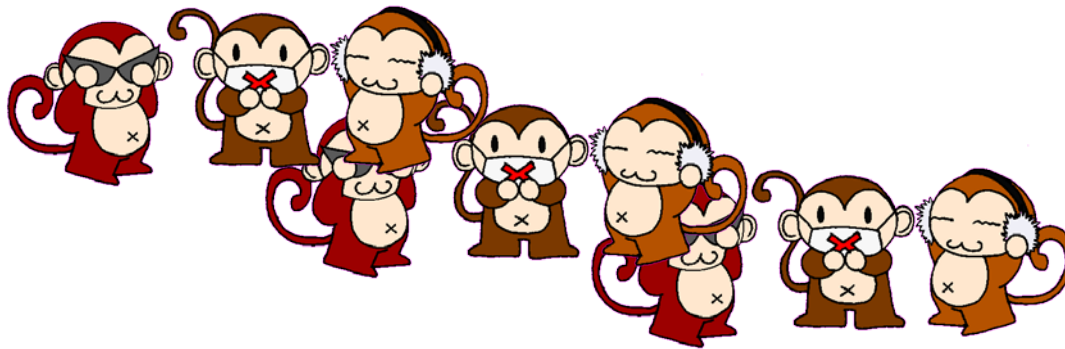
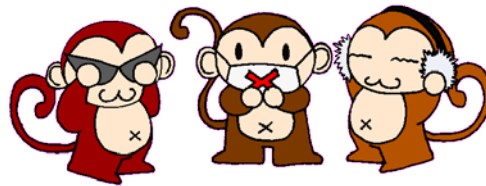
黒？



- 画像元) [http://mitemitei.com/h/\(print\)031.htm](http://mitemitei.com/h/(print)031.htm)

# 実は。

- 透過GIFでした。
  - (そして JPEG は透明色を持ってない)



# 画像入出力サンプル(take2)

- 背景色を設定してみた。(けど GIF はダメ?)

GIF

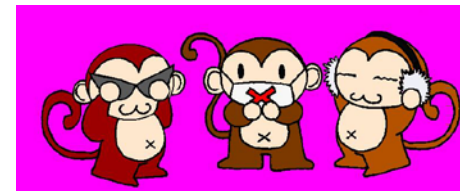
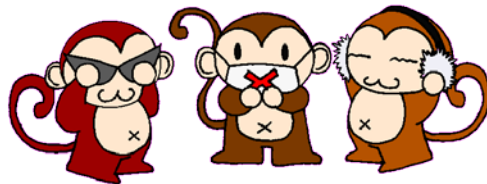


JPEG

```
<php
$im = imagecreatefromgif($argv[1]);
$white = imagecolorallocate($im, 255, 255, 255);
imagecolortransparent($im, $white);
imagejpeg($im);
```

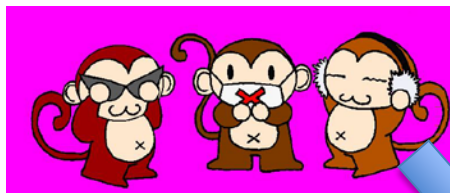
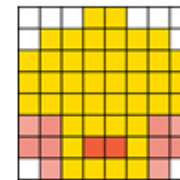
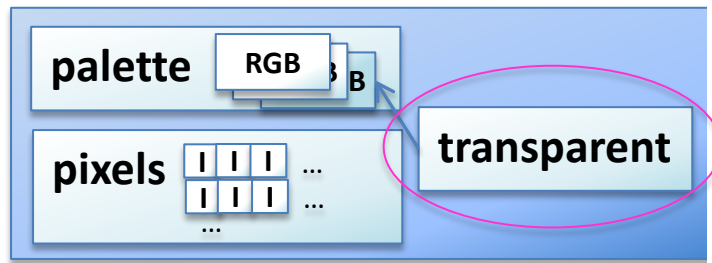
```
% php gif2jpeg.php tomo_pr_263.gif > tomo_pr_263_white.jpg
```

紫?

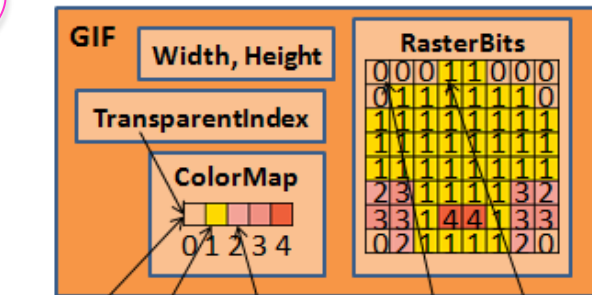
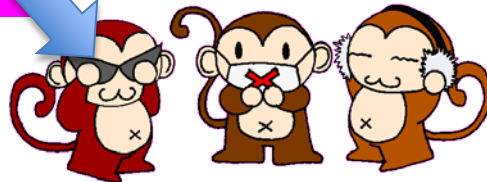


# GIF の透過

- 透明度のないpalette の色の中から、どれか一つを透明色にする。(透明色は後付け)



背景を紫で作って  
後で透明化



00 00 00 fe d9 00 f3 a3 98 ... 00 00 00 01 01 00 00 00  
←---→ ←---→ ←---→ 00 01 01 01 01 01 01 00  
透明 黄 肌色 01 01 01 01 01 01 01 01  
...

# 画像入出力サンプル(take3)

- パレットの色を入れ替えてみた

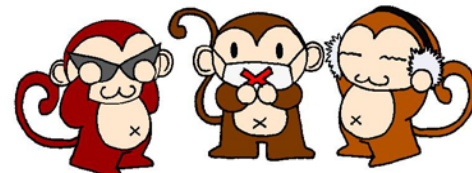
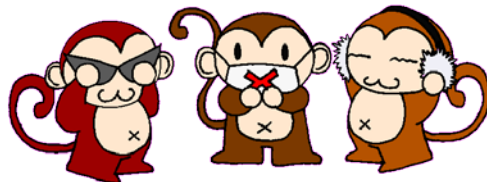
GIF



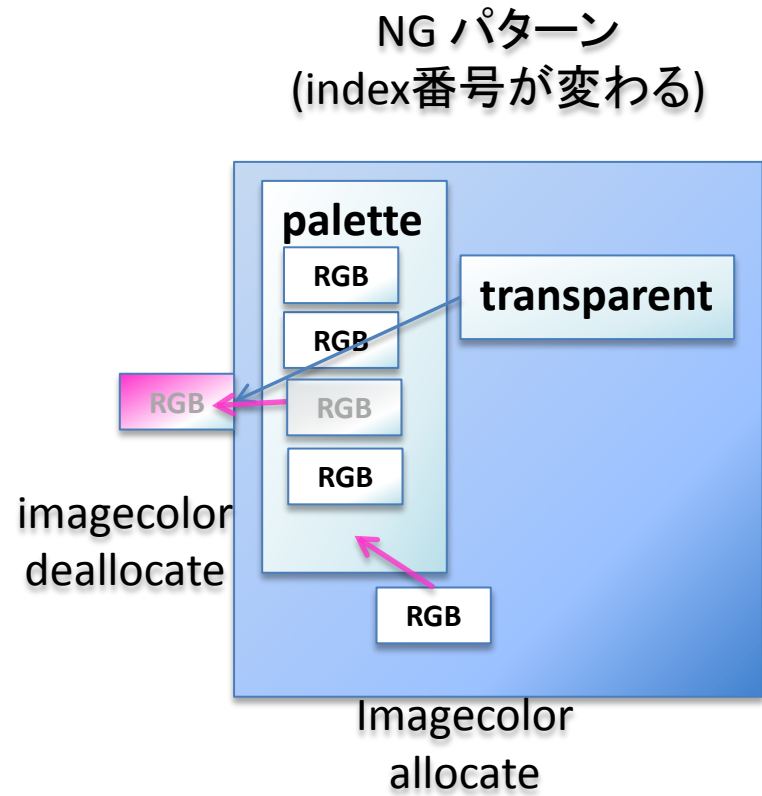
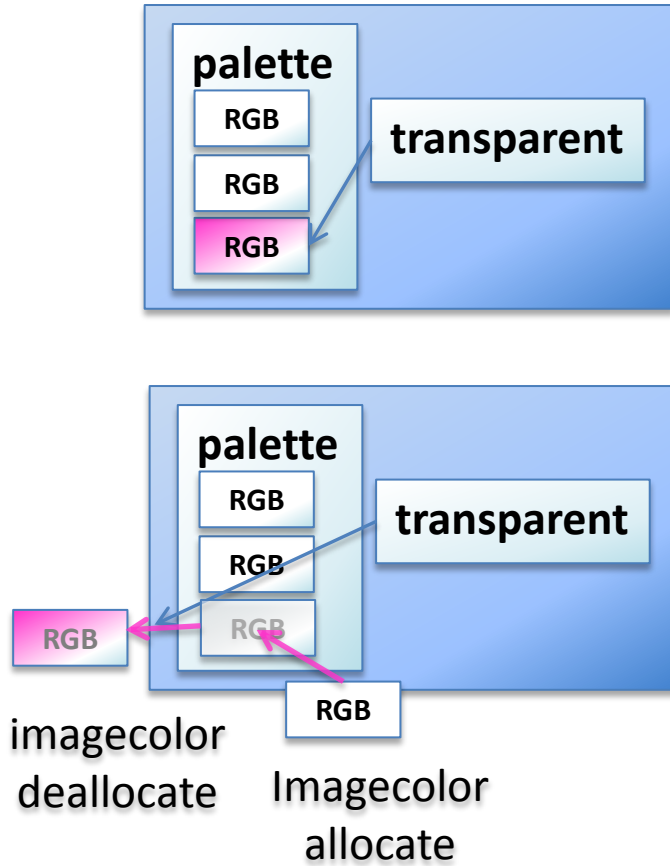
JPEG

```
<?php
$im = imagecreatefromgif($argv[1]);
$trans = imagecolortransparent($im);
if ($trans != -1) {
    imagecolordeallocate($im, $trans);
    $trans2 = imagecolorallocate($im, 255, 255, 255);
}
imagejpeg($im);
```

```
% php gif2jpeg.php tomo_pr_263.gif > tomo_pr_263_white.jpg
```



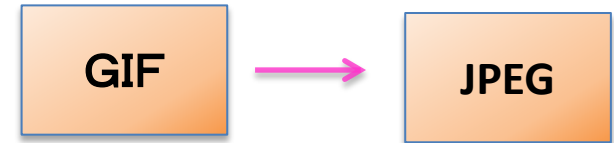
# 何をしたのか？



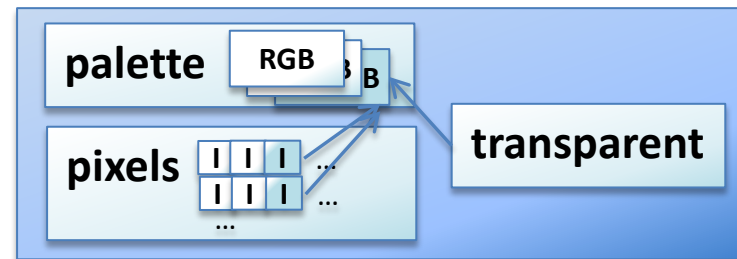


# 画像入出力サンプル(take4)

- ピクセル上書き



```
<?php
$im = imagecreatefromgif($argv[1]);
$trans1 = imagecolortransparent($im);
if ($trans1 != -1) {
    imagecolordeallocate($im, $trans1);
    $trans2 = imagecolorallocate($im, 255, 255, 255);
    if ($trans1 != $trans2) {
        for ($y = 0 ; $y < imagesy($im) ; $y++) {
            for ($x = 0 ; $x < imagesx($im) ; $x++) {
                if (imagecolorat($im, $x, $y) === $trans1) {
                    imagesetpixel($im, $x, $y, $trans2);
                }
            }
        }
    }
}
}
imagejpeg($im);
```



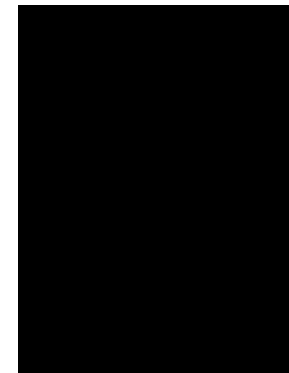
# 画像生成サンプル

- 真っ黒な画像

```
<?php
$im = imagecreate(240, 320);
$black = imagecolorallocate($im, 0, 0, 0);
imagefill($im, 0, 0, $black);

imagepng($im);
```

```
% php pureblack.php > pureblack.png
```

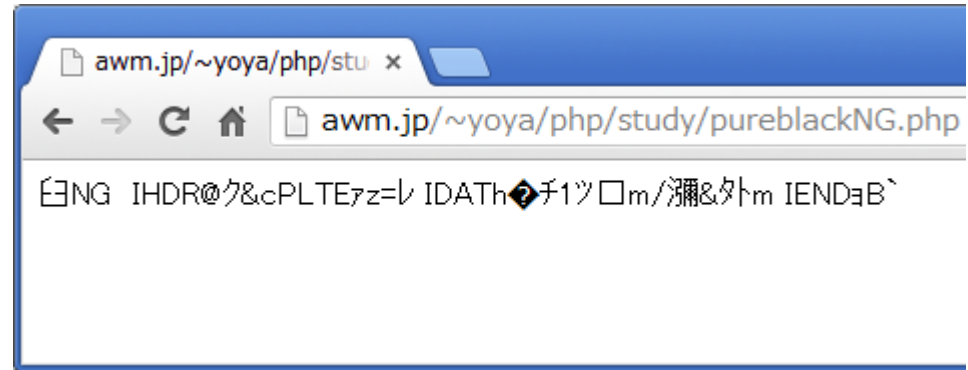


# 画像生成サンプル (Web版)

- 真っ黒な画像

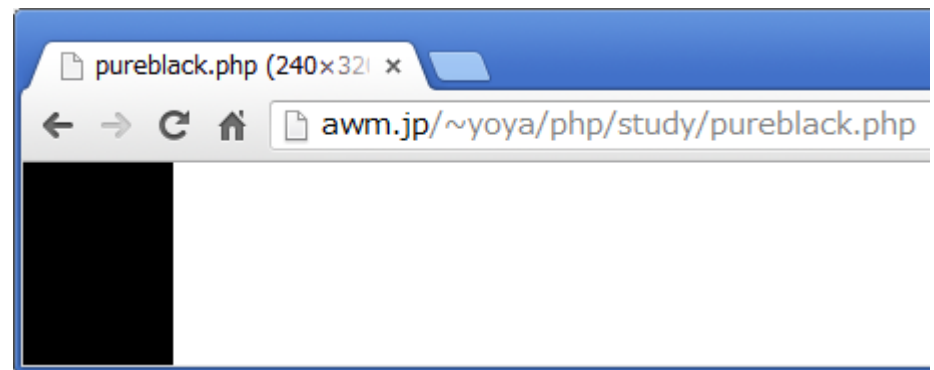
```
<?php
$im = imagecreate(240, 320);
$black = imagecolorallocate($im, 0, 0, 0);
imagefill($im, 0, 0, $black);

imagepng($im);
```



```
<?php
$im = imagecreate(240, 320);
$black = imagecolorallocate($im, 0, 0, 0);
imagefill($im, 0, 0, $black);

header('Content-type: image/png');
imagepng($im);
```



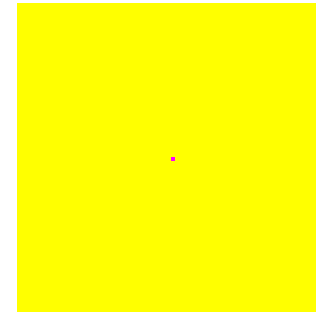
# GD の流儀

- 色は `imagecolorallocate` 経由で使う

```
$im = imagecreate(100, 100);  
$b = imagecolorallocate($im, 255, 255, 0);  
imagefill($im, 0, 0, $b);  
$c = imagecolorallocate($im, 255, 0, 255);  
imagesetpixel($im, 50, 50, $c);  
imagepng($im);
```

$\$c = 0 \sim 255$

Index  
↔  
1byte



- true color でも `allocate` 経由

```
$im = imagecreatetruecolor(240, 320);  
$b = imagecolorallocate($im, 255, 255, 0);  
imagefill($im, 0, 0, $b);  
$c = imagecolorallocate($im, 255, 0, 255);  
imagesetpixel($im, 50, 50, $c);  
imagepng($im);
```

```
// $c = imagecolorallocate($im, 255, 0, 255);  
$c = (255 << 16) + (255 < 8) + 0;
```

$\$c = 0 \sim 2147483647$   
(=0x7FFFFFFF)  
Index  
↔  
4 bytes

0~127 0~255 0~255 0~255  
Alpha Red Green Blue  
↔ ↔ ↔ ↔  
1byte 1byte 1byte 1byte

# GD マニュアル

- 後は、これを見れば大体分かる

– <http://www.php.net/manual/ja/ref.image.php>



The screenshot shows the PHP GD manual page for the `imagecopymerge` function. The page title is «imagecopymerge による半透明なすかしの作成». The page is in English and was last updated on Friday, 21 Jun 2013. The table of contents for the GD and Image functions section is as follows:

- [gd\\_info](#) — 現在インストールされているGDライブラリに関する情報を取得する
- [getimagesize](#) — 画像の大きさを取得する
- [getimagesizefromstring](#) — 画像の大きさを文字列から取得する
- [image\\_type\\_to\\_extension](#) — 画像形式からファイルの拡張子を取得する
- [image\\_type\\_to\\_mime\\_type](#) — `getimagesize`, `exif_read_data`, `exif_thumbnail`, `exif_imagetype`から返される 画像形式のMIMEタイプを取得する
- [image2wbmp](#) — 画像をブラウザあるいはファイルに出力する
- [imageaffine](#) — 元の画像を、オプションのクリッピング領域を使ってアフィン変換した画像を返す

# 応用例

- 色んな形式の画像ファイルを生成
- 色の数をカウントする
- 画像差分抽出
- ドット絵っぽい変換
- 色分布の3D表示

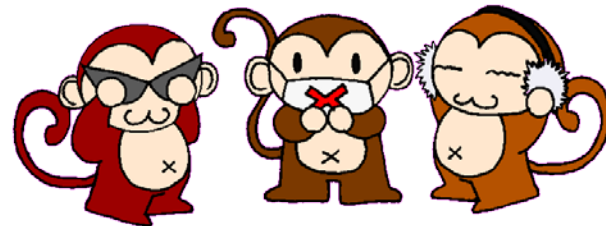
# 色々な形式の画像ファイルを生成

- ビットマップ画像フォーマット毎のテスト素材
  - <http://d.hatena.ne.jp/yoya/20110622/gd>
- PNG/JPEG/GIF
- パレット形式/トゥルーカラー形式
- 透明度なし/あり

# 色の数をカウントする

- colorcount.php
  - <http://d.hatena.ne.jp/yoya/20120421/php>

```
$im = imagecreatefromstring(file_get_contents($argv[1]));  
$width = imagesx($im); $height = imagesy($im);  
$colorcount = array();  
for ($x = 0; $x < $width; $x++){  
  for ($y = 0; $y < $height; $y++){  
    $colorindex = imagecolorat($im, $x, $y);  
    $colorcount[$colorindex]++;  
  }  
}  
echo count($colorcount).PHP_EOL;
```



```
% php colorcount.php tomo_pr_263.gif  
12
```



# 色の差分抽出

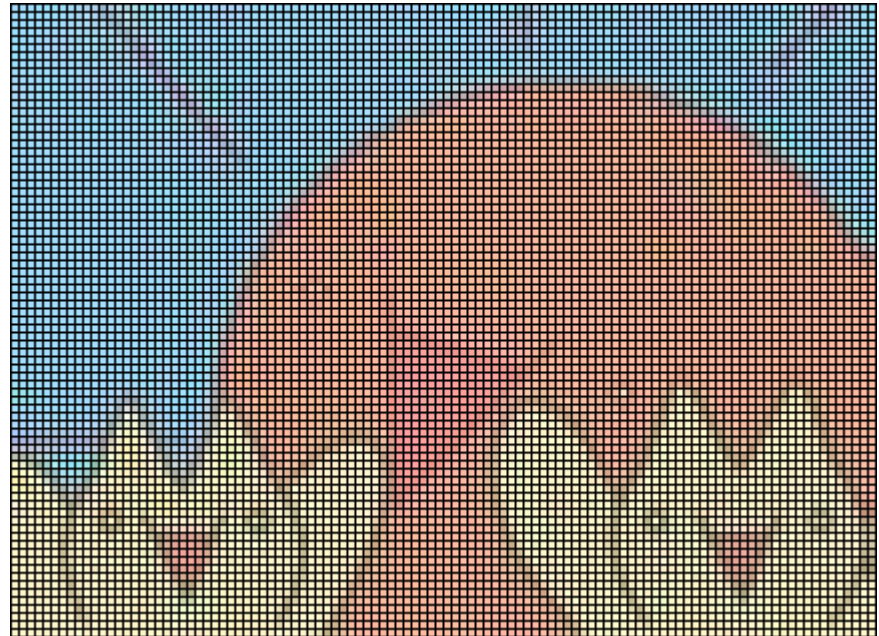
- PHP で画像比較
  - <http://d.hatena.ne.jp/yoya/20120712/php>
- 減色等でどのように画像が変わったか？

```
% convert aria.jpg aria.gif  
% php bitmap_diff.php aria.jpg aria.gif aria_diff.png
```



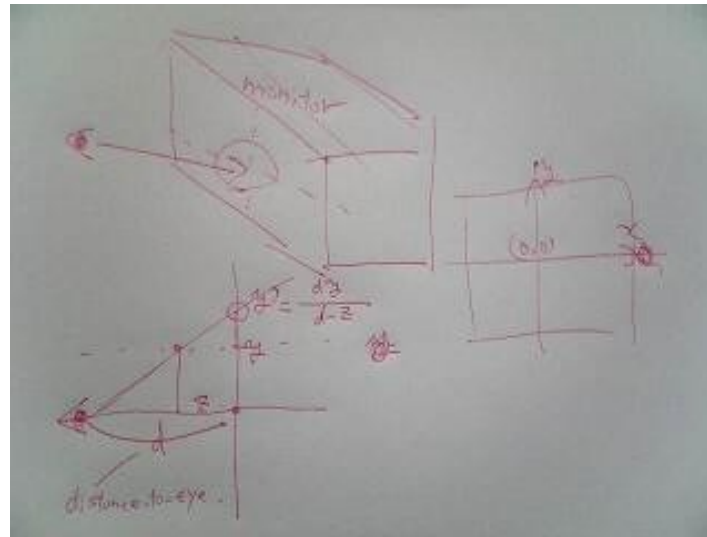
# ドット絵っぽい変換

- ドット絵っぽく見せる変換ツール
  - <https://github.com/yoya/misc/blob/master/php/dottize.php>
- <http://awm.jp/~yoya/php/image/dottize.php>



# 色分布の3D表示 (1/2)

- PHP で 3D plot
  - <http://d.hatena.ne.jp/yoya/20080925>

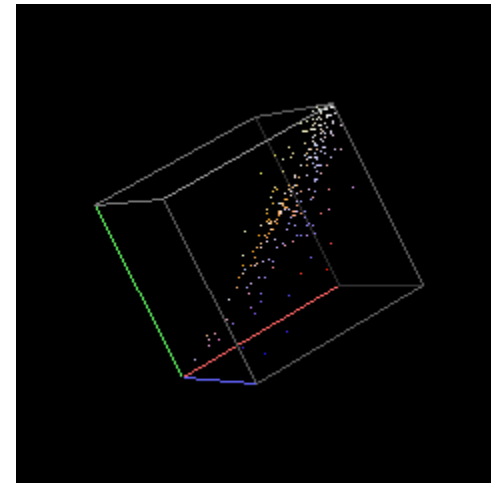


- <http://awm.jp/~yoya/php/3d/y3d.phps>

# 色分布の3D表示 (2/2)

- GIF画像の色分布

- <http://d.hatena.ne.jp/yoya/20080927>



- <http://awm.jp/~yoya/php/>

# その他

- 文字を埋め込めます
- ブラシ指定で書き込めます。タイル状に？
- 描画図形のプリミティブ色々対応してます
  - 線だけでなく円とか多角形とか。

# GD の限界

- 8bit depth しか扱えない。(16bit カラーを扱い事が結構ある) ImageMagick の内部形式が大抵16bit なので
- 透明度が 0~127 なので精度が半分
- ピクセルの操作が遠回り (パレットインデックス経由)
- サムネールが汚いらしい。減色も汚い
- 処理が重たいらしい。(Klabさんが高速化サービスを出す位には重たい)

# PHP5.5

- PHP 5.5 で増えた GD functions

- <http://d.hatena.ne.jp/yoya/20130621/gd>

imagepalettetotruecolor

imageflip

imagecrop, imagecropauto

imageaffine, imageaffinematrixget, imageaffinematrixconcat

imagescale

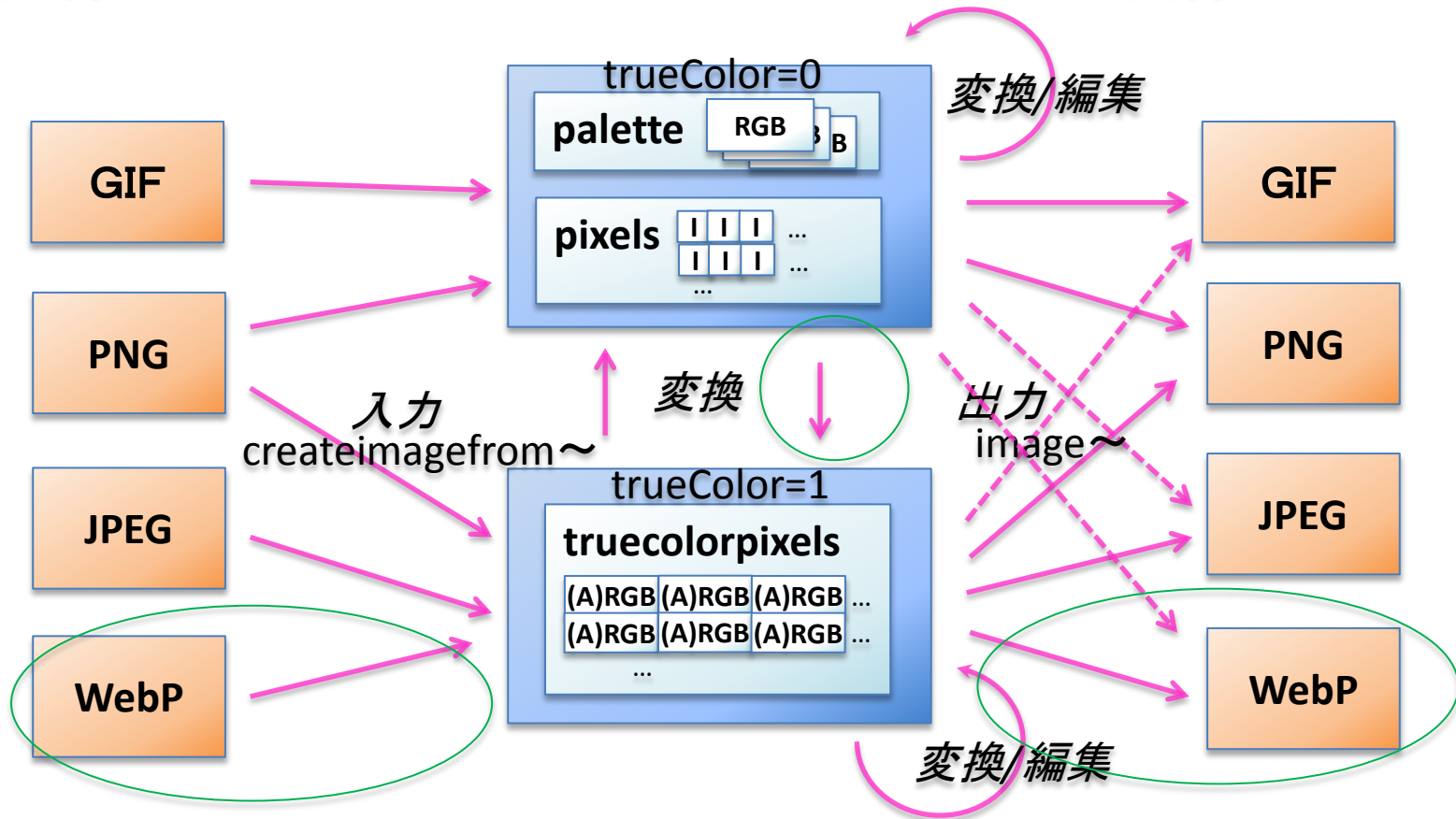
imagesetinterpolation

# 入出力フロー (PHP5.5)

画像ファイル

GD リソース

画像ファイル

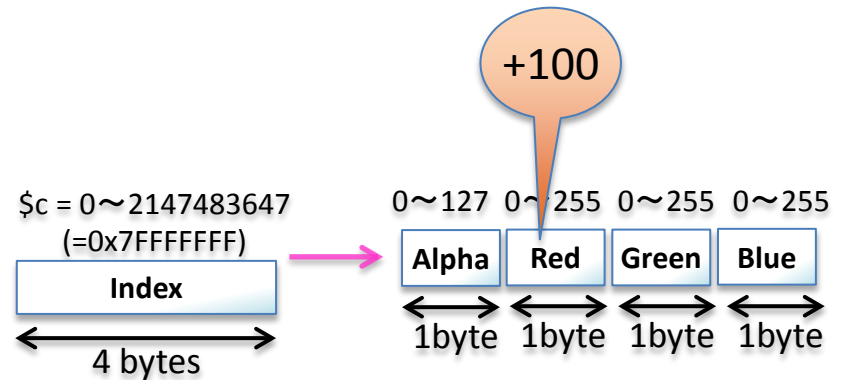




# imagepalettetotruecolor

- imagered.php

```
$imgdata = file_get_contents($argv[1]);  
$im = imagecreatefromstring($imgdata);  
imagepalettetotruecolor($im);  
for ($y = 0 ; $y < imagesy($im) ; $y++) {  
  for ($x = 0 ; $x < imagesx($im) ; $x++) {  
    $c = imagecolorat($im, $x, $y);  
    $r = ($c >> 16) & 0xFF;  
    $r2 = $r + 100;  
    $r2 = ($r2 < 256)?$r:255;  
    $c2 = ($c & 0x7F00FFFF) + $r2 << 16;  
    imagesetpixel($im, $x, $y, $c2);  
  }  
}  
  
imagepng($im);
```

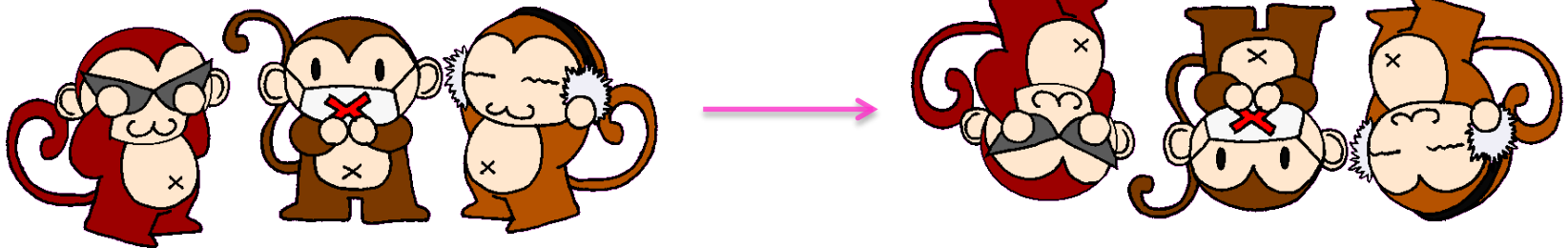


# imageflip

- 縦、横、又は両方で画像をひっくり返す

```
$imgData = file_get_contents($argv[1]);  
$im = imagecreatefromstring($imgData);  
imageflip($im, IMG_FLIP_VERTICAL);  
echo imagegif($im);
```

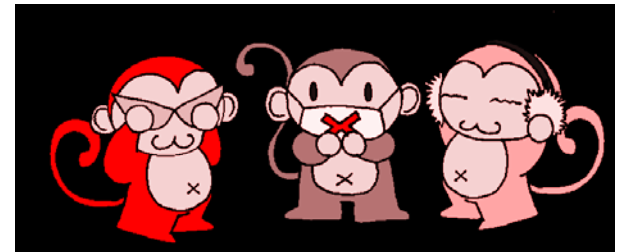
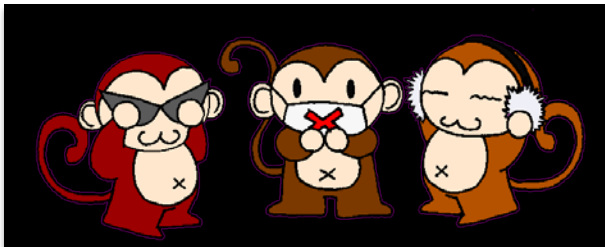
```
% php imageflip.php tomo_pr_263.gif > tomo_pr_263_flip.gif
```



# Imagered.php

- カラー効果で赤に +100

```
% php imagered.php tomo_pr_263.gif > tomo_pr_263_red.gif
```



以上です

ご清聴ありがとうございました