

PHP版 バーコード作成ライブラリ

Barcode.php

説明書

Version 4.0

2025年

Pao@Office

はじめに

Barcode.php は、PHP環境下で動作する、バーコード作成ツール(クラス群)の総称です。

Barcode.php は、次のことを念頭において開発いたしました。

1. 高い精度

RJSのバーコード検査機、Model L2000を使用してバーコードの精度を細かく検査しています。このため、他のバーコード作成ツールに比べても高精度なバーコードを作成できます。また、プリンタによる線のにじみや読み取り精度の低下にも対応し、バーコードの線幅微調整も自在に行えます。

2. 使いやすさ

直感的なインターフェースを提供しています。マニュアルの使用例に示されている通り、2~3つのステップでバーコードの出力が可能です。

3. 軽量

「Barcode.php」はシステムへの負荷が極めて小さく、わずか数MBのメモリで動作します。

4. 汎用性

GDオブジェクトに対してバーコードを描画する仕組みになっております。そのため、バーコードを作成するアプリケーションから画像ファイル等さまざまな用途で利用することができます。PDFへの出力にも容易かつ高精度なバーコードを生成できます。また、プリンタによるバーコード読み取り精度の微調整も柔軟に行えます。PDF出力やブラウザ出力に関しては、すぐに試用できるサンプルプログラムも提供しています。

「Barcode.php」を使用することで、PHPを活用したシステム開発において、簡単かつ便利にバーコードの出力を行うことができます。

5. 柔軟な描画オプション (Ver 4.0 新機能)

モジュール幅指定のdraw()に加え、全体幅指定のdraw_by_width()に対応。FitWidthプロパティにより、幅ピッタリ描画とドット精度優先描画を選択できます。PNG/JPEG/GIF/BMPに加え、SVGベクター出力にも対応しています。

2025年 作者

目次

1. Barcode.phpの動作環境・インストール方法

- 1-1. [動作環境](#)
- 1-2. [インストール方法](#)

2. Barcode.phpの機能

- 2-1. [機能概要](#)
 - 2-1-1. [一次元バーコードの種類](#)
 - 2-1-2. [二次元バーコードの種類](#)
 - 2-1-3. [GS1 Databar \(RSS\) の種類](#)
 - 2-1-4. [Ver 4.0.0 新機能](#)
- 2-2. [一次元バーコード作成クラスの機能](#)
- 2-3. [郵便カスタマバーコード作成クラスの機能](#)
- 2-4. [QRコード作成クラスの機能](#)
- 2-5. [DataMatrix作成クラスの機能](#)
- 2-6. [PDF417作成クラスの機能](#)
- 2-7. [GS1データバー\(RSS\)作成クラスの機能](#)

3. アプリケーションプログラムからBarcode.phpの使用方法

- 3-1. [クラス仕様](#)
 - 3-1-1. [概要](#)
 - 3-1-2. [一次元バーコードクラスメンバ](#)
 - 3-1-3. [GS1データバー\(RSS\)クラスメンバ](#)
 - 3-1-4. [郵便カスタマバーコードクラスメンバ](#)
 - 3-1-5. [QRコードクラスメンバ](#)
 - 3-1-6. [DataMatrixクラスメンバ](#)
 - 3-1-7. [Pdf417クラスメンバ](#)
- 3-2. [使用例・サンプルプログラム](#)

4. 使用条件等

- 4-1. [試用版と製品版](#)
- 4-2. [使用許諾](#)
- 4-3. [代金支払い方法\(ユーザ登録の方法\)](#)

1. Barcode.phpの動作環境・インストール方法

1-1. 動作環境

- PHP 5.6 / PHP 7.x / PHP 8.x (8.5まで動作確認済み) が動作する環境であること。
- PHP環境として、GDが利用可能であること。
- SVG出力のみを使用する場合は、GDは不要です。

1-2. インストール方法

以下のPHPソースファイルをコピーしてお使いください。

○ 共通ルーチン → 必須

- Common.php
- CheckDigit.php

○ 各バーコード描画 → お使いのバーコードのみ必要

- Code39.php
- Code93.php
- Code128.php
- EAN128.php (GS1-128)
- ITF.php
- Jan13.php
- Jan8.php
- Matrix2of5.php
- NEC2of5.php
- NW7.php
- YubinCustomer.php
- QR.php
- DataMatrix.php + DataMatrixディレクトリ
- Pdf417.php
- Databar14.php
- DatabarLimited.php
- DatabarExpanded.php

試用版：以下のURLより、Docker版またはVMWare Playerイメージ版をダウンロードしてお試しください。

<http://www.pao.ac/barcode.php/>

Docker版（推奨）

- CentOS Stream 9 + PHP 8.3
- Ubuntu 24.04 LTS + PHP 8.5
- Debian 9 + PHP 5.6 (旧版)

VMWare版（OSイメージごとなので大きいですが自由度はあります）

- Ubuntu 24.04 LTS + PHP 8.3

Barcode.phpのサンプルプログラムは、以下のいずれかのディレクトリにございます。

- /var/www/html/sample
- /usr/local/apache2/htdocs/sample

製品版：メールで製品PHPソースファイルのURLをお送りします。zipファイルをダウンロードしてを解凍後、任意のディレクトリにコピーしてお使いください。

プログラムからは、以下のように記述してお使いください。

```
require_once("Code39.php");
```

2. Barcode.phpの機能

2-1. 機能概要

Barcode.php は、以下のバーコードの作成が可能です。

- (1) JAN-13(EAN-13)
- (2) JAN-8(EAN-8)
- (3) UPC-A
- (4) UPC-E
- (5) ITF(インターリード 2 of 5)
- (6) Matrix 2 of 5
- (7) NEC 2 of 5 (Coop 2 of 5)
- (8) NW-7(CODA-BAR)
- (9) CODE39
- (10) CODE93
- (11) CODE128
- (12) GS1-128 (UCC/EAN-128)

- コンビニ向け標準料金代理収納用バーコード
- 医療用 医薬品等のバーコード
- 医療用 医療材料等のバーコード
- 食肉標準物流バーコード「基本バーコード」

- (13) 郵便カスタマバーコード
- (14) GS1 Databar 標準型 (RSS-14) ... ver 3.0 において追加
- (15) GS1 Databar 限定型 (RSS Limited) ... ver 3.0 において追加
- (16) GS1 Databar 拡張型 (RSS Expanded) ... ver 3.0 において追加
- (17) QRコード
- (18) 標準料金代理収納用バーコード(コンビニバーコード)
- (19) DataMatrix (GS1 DataMatrix)
- (20) PDF417

※郵便カスタマバーコード・GS1 Databar・二次元バーコード(QR / DataMatrix / PDF417)以外は、以降総称して「一次元バーコード」と呼びます。

Barcode.php では、上記の各バーコードを作成するために、バーコードの種類ごとに全て別々のクラスとして利用することができます。

Barcode.php の各バーコード作成クラスは、バーコードを描画する `draw` メソッドを実装しています。Draw メソッドの戻り値は、バーコードが描画された GD ライブラリの画像(image)オブジェクトです。

Ver 4.0.0 新機能 : OutputSVG プロパティを `true` に設定すると、戻り値はSVG形式の文字列になります。

2-1-1. 一次元バーコードの種類

次の種類のバーコードを出力できます。

1. JAN-13(EAN-13)

数字13桁。12桁指定時、13桁目のチェックディジットを自動計算付与。

2. JAN-8(EAN-8)

数字8桁。7桁指定時、8桁目のチェックディジットを自動計算付与。

3. UPC-A

数字12桁。11桁指定時、12桁目のチェックディジットを自動計算付与。

4. UPC-E

数字7桁。6桁指定時、7桁目のチェックディジットを自動計算付与。

5. ITF(インターリード 2 of 5)

使用可能文字：数字

6. Matrix 2 of 5

使用可能文字：数字

7. NEC 2 of 5 (Coop 2 of 5)

使用可能文字：数字

8. NW-7(CODA-BAR)

使用可能文字：ABCD.+;/-\$-0123456789

1文字目 ABC いずれか入力した文字がスタート・ストップキャラクタとなる。入力がない場合 C が既定値。

パラメータ：スタート・ストップキャラクタ表示／非表示

9. CODE39

使用可能文字 : 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-. *\$/+%

パラメータ : スタート・ストップキャラクタ表示／非表示

10. CODE93

使用可能文字 : 1234567890ABCDEFGHIJKLMNOPQRSTUVWXYZ-. *\$/+%

11. CODE128

多くの半角英数文字記号・エスケープシーケンスが使用可能。

CODE A/B/C により出力文字・パターンを切り替えることが可能。

パラメータ : どのコードパターンを使用するか。 AUTO / CODE_A / CODE_B / CODE_C 既定値は、 AUTO。 AUTOは、 数字4文字続ければ CODE C にコードチェンジするなど、 バーコードが一番小さくなるように自動調整する。

12. GS1-128 (UCC/EAN-128)

- コンビニ向け標準料金代理収納用バーコード
- 医療用 医薬品等のバーコード
- 医療用 医療材料等のバーコード
- 食肉標準物流バーコード「基本バーコード」

コンビニバーコード 入力例 : {FNC1}9191234500000000000004520875004013100295006

13. 郵便力スタマバーコード

郵便番号 + 住所の英数字部分のみ入力

例 : 27500263-29-2-401

パラメータ : 8~11.5 ポイント(大きさ)

2-1-2. 二次元バーコードの種類

次の種類のバーコードを出力できます。

1. QRコード

パラメータ : (既定値は下線)

(1) エラー訂正レベル : L / M / Q / H (2) 文字種 : 数字 / 大文字英数字 / 8ビットバイトデータ(漢字含む) (3) バージョン : 1~40 5 (4) 全角文字エンコード : "shift-jis" / "utf8" / etc..

2. DataMatrix (GS1 DataMatrix)

パラメータ : (既定値は下線)

(1) シンボルコードサイズ : 自動 / 10x10~144x144 / 8x18~16x48 等 (2) 全角文字エンコード : "utf8" / "shift-jis" / etc..

3. PDF417

パラメータ : (既定値は下線)

(1) エラー訂正レベル自動決定 : する / しない (2) エラー訂正レベル : 1~8 2 (3) アスペクト比 : 0.5 / 1.0 / 2.0 等 (4) 列・行数決定方法 : アスペクト比より自動決定 / 列数指定 / 行数指定 / 列数行数指定 (5) 指定列数 : 1 ~30 5 (6) 指定行数 : 3~90 5 (7) 全角文字エンコード : "shift-jis" / "utf8" / etc..

2-1-3. GS1 Databar (RSS) の種類

次の種類のバーコードを出力できます。

1. GS1データバーOmni-directional / 表示桁数：数字14桁(GTIN)

Omni-directional(標準型)

高さが狭い Truncated(カット型)という種類もございますが、この標準型の高さを調整して出力してください。

Stacked (2層型)

Stacked Omni-directional (標準2層型)

2. GS1データバーLimited / 表示桁数：数字14桁(GTIN)

Limited(限定型)

一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。

3. GS1データバーExpanded / 表示桁数：最大数字74桁または英字41文字

Expanded(拡張一層型)

一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。

Stacked Expanded(拡張多層型)

このバーコードは、各段(層)の割合は同一な仕様です。段と段(層と層)間の小さな3段の正方形の高さを固定し、ユーザより引数で指定された高さにより、各段(層)がその割合によって同じ高さで伸び縮みします。

2-1-4. Ver 4.0.0 新機能

Barcode.php Ver 4.0.0 では、以下の新機能が追加されました。

(1) 幅指定描画 (draw_by_width)

従来の draw メソッドに加え、バーコード全体の幅を指定して描画できる draw_by_width メソッドを追加しました。これにより、レイアウトに合わせた柔軟なバーコード生成が可能になります。

```
// 従来の描画方法（モジュール幅指定）
$img = $barcode->draw($code, 2, 50);

// 新しい描画方法（全体幅指定）
$img = $barcode->draw_by_width($code, 300, 50);
```

(2) 幅ピッタリ／ドットピッタリ選択 (FitWidth プロパティ)

FitWidth プロパティにより、描画精度の制御が可能になりました。

FitWidth	動作	用途
true (既定値)	指定幅に完全フィット（小数モジュール）	見た目・レイアウト重視
false	整数ドット単位で描画	スキャナ読み取り精度重視

```
// 幅ピッタリモード（既定値）
$barcode->FitWidth = true;

// ドットピッタリモード
$barcode->FitWidth = false;
```

(3) SVG出力対応

ベクター形式のSVG出力に対応しました。拡大縮小しても劣化しない高品質なバーコードを生成できます。

```
// SVG出力モードを有効化
$barcode->OutputSVG = true;

// SVG文字列を取得
$svg = $barcode->draw($code, 2, 50);

// ファイルに保存
file_put_contents("barcode.svg", $svg);
```

SVG出力では、戻り値がGD画像オブジェクトではなく、SVG形式の文字列となります。

2-2. 一次元バーコード作成クラスの機能

Barcode.php の各一次元バーコード作成クラスは、以下の機能を有します。

(1) バーコードの描画

コード、横方向の最少描画ドット数、バーコードの高さを指定してバーコードを描画します。

少し特殊な GS1-128(UCC/EAN-128)において、AI(アプリケーション識別子)挿入方法は2通りございます。

(1) 可変長項目(データブロック)の後の AI には、FNC1 を挿入 ⇒これまで通り "{FNC1}" を付ける。例："{FNC1}21" のようにコードを指定

(2) 固定長項目(データブロック)の後の AI には、固定長のため目印の FNC1 は不要 ⇒新しく追加した "{AI}" を付ける。例："{AI}21" のようにコードを指定

"{AI}" を指定して FNC1 を挿入しない場合も、カッコ()付コード文字は出力されます。例えば入力コードに "{AI}21" を指定した場合、添え字には(21)と出力されます。

例) (01)04512345670016(21)1 ⇒(01)の前には FNC1 を挿入し(21)の前には挿入しない。コード指定方法 → "{FNC1}0104512345670016{AI}211"

(2) 幅指定によるバーコードの描画 (Ver 4.0 新機能)

コード、バーコード全体の幅、バーコードの高さを指定してバーコードを描画します。

(3) 添字の描画

バーコードの下にコードの文字列を描画します(既定値)。

プロパティの設定で描画をしないようにすることも可能です。

添字のフォントをプロパティで指定することも可能です。

CODE39/NW-7(CODABAR) のみスタート・ストップキャラクタを印字するかどうかをプロパティで指定することができます。既定値は印字しません。

(4) SVG出力 (Ver 4.0 新機能)

OutputSVG プロパティを true に設定すると、draw / draw_by_width メソッドの戻り値がSVG文字列になります。

2-3. 郵便カスタマバーコード作成クラスの機能

Barcode.php の郵便カスタマバーコード作成クラスは、以下の機能を有します。

(1) バーコードの描画

コード、サイズを指定してバーコードを描画します。

コードの表記は・・・ [郵便番号の数字部分7桁]+[郵便番号では不明部分の住所の英数字を「-」区切り] で、指定してください。

例) 〒116-0013 東京都荒川区西日暮里五丁目37番5号スタートアップオフィスA-207号室 コード指定方法
→ 「11600135-37-5-A-207」

※詳しくは、旧郵政省のwebページにマニュアルがございますのでご覧になってください。

2-4. QRコード作成クラスの機能

Barcode.php の QRコード作成クラスは、以下の機能を有します。

バーコードの描画

コードと最少描画ドット数、又は、コードと幅を指定して描画することが可能です。

プロパティで以下の項目を指定することが必要です。

- バージョン(1~40)
- エラー訂正レベル(L,M,Q,H)

SVG出力 (Ver 4.0 新機能)

OutputSVG プロパティを `true` に設定すると、`draw` / `draw_by_width` メソッドの戻り値がSVG文字列になります。

2-5. DataMatrix作成クラスの機能

Barcode.php の DataMatrix 作成クラスは、以下の機能を有します。

DataMatrixの描画

コードと最少描画ドット数、又は、コードと幅を指定して描画することが可能です。

プロパティで以下の項目を指定することができます。

- シンボルコードサイズ 既定値 : DxCodeSize.DxSzAuto

enum DxCodeSize

DxSzRectAuto

DxSzAuto

DxSzShapeAuto

DxSz10x10

DxSz12x12

DxSz14x14

DxSz16x16

DxSz18x18

DxSz20x20

DxSz22x22

DxSz24x24

DxSz26x26

DxSz32x32

DxSz36x36

DxSz40x40

DxSz44x44

DxSz48x48

DxSz52x52

DxSz64x64

DxSz72x72

DxSz80x80

DxSz88x88

enum DxCodeSize

DxSz96x96

DxSz104x104

DxSz120x120

DxSz132x132

DxSz144x144

DxSz12x36

DxSz16x36

DxSz16x48

- 全角エンコーディング ("utf-8", "shift-jis", etc..) 既定値 : utf-8

SVG出力 (Ver 4.0 新機能)

OutputSVG プロパティを true に設定すると、draw / draw_width メソッドの戻り値がSVG文字列になります。

2-6. PDF417作成クラスの機能

Barcode.php の PDF417 作成クラスは、以下の機能を有します。

PDF417の描画

コードと最少描画ドット数を指定して描画することが可能です。

プロパティで以下の項目を指定することができます。

- **サイズ種別** ...データ列数・行数決定方法
 - 自動 Pdf417.SIZE_KIND.AUTO (既定値)
 - データ列数指定 Pdf417.SIZE_KIND.COLUMNS
 - データ行数指定 Pdf417.SIZE_KIND.ROWS
 - データ列数・行数指定 Pdf417.SIZE_KIND.COLUMNS_AND_ROWS
- **行数** ...出力データ行数指定 サイズ種別が、出力行数指定の場合 = (自動サイズでない・列数指定でない場合)有効 3~90 既定値 : 5
- **列数** ...出力データカラム数指定 サイズ種別が、出力列数指定の場合 = (自動サイズでない・行数指定でない場合)有効 1~30
- **エラーレベル** 0~8 既定値 : 2
- **エラーレベル自動決定** 自動でエラー訂正レベルを決定(する・しない) 既定値 : true(する)
- **縦横アクセプト比** シンボルの縦横比、シンボルレアスペクトレシオ (比) 既定値 : 0.5
- **全角エンコーディング** ("utf-8", "shift-jis", etc..) 既定値 : utf-8

幅指定による描画 (Ver 4.0 新機能)

draw_by_width メソッドにより、PDF417全体の幅を指定して描画することが可能です。

SVG出力 (Ver 4.0 新機能)

OutputSVG プロパティを true に設定すると、draw / draw_by_width メソッドの戻り値がSVG文字列になります。

2-7. GS1データバー(RSS)作成クラスの機能

Barcode.php の各GS1データバー(RSS)作成クラスは、以下の機能を有します。

(1) バーコードの描画

コード、横方向の最少描画ドット数、バーコードの高さを指定してバーコードを描画します。

(2) 幅指定によるバーコードの描画 (Ver 4.0 新機能)

コード、バーコード全体の幅、バーコードの高さを指定してバーコードを描画します。

(3) バーコードの高さ指定とその出力結果

高さ指定とその出力結果について Barcode.php 独自の仕様がございます。

1. GS1データバーOmni-directional / 表示桁数：数字14桁(GTIN)

Omni-directional (標準型) 一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。

Stacked (2層型) バーコード上段・センター部分・下段の高さ割合が決まっているため、高さの指定があつても割合をキープして出力することが基本です。

Barcode.php では、DrawDirect / DrawDelicate メソッドを使用した時に、センター部分を含む3段の割合を確実にキープします。

Draw メソッドの場合は、センターの小さな正方形群の各正方形割合はキープしながら、ユーザより引数で指定された高さにより上段・下段の割合に従って上段下段の高さを決定します。従って Draw メソッドでは、本来決まっているセンター部分を含む3段の高さ割合は保たれません。指定された高さにより、上・下段が決められている割合を保ちながら伸び縮みします。

Stacked Omni-directional (標準2層型) バーコードの上段・下段の割合は同一で、センター部分(小さな正方形3段)がある仕様のバーコードです。従ってセンター部分の高さを固定し、ユーザより引数で指定された高さにより、上・下段がその割合によって同じ高さで伸び縮みします。

2. GS1データバーLimited / 表示桁数：数字14桁(GTIN)

Limited (限定型) 一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。

3. GS1データバーExpanded / 表示桁数：最大数字74桁または英字41文字

Expanded(拡張一層型) 一次元バーコードなので、幅高さは通常の一次元バーコードと同様に自在です。

Stacked Expanded(拡張多層型) このバーコードは、各段(層)の割合は同一な仕様です。段と段(層と層)間の小さな3段の正方形の高さを固定し、ユーザより引数で指定された高さにより、各段(層)がその割合によって同じ高さで伸び縮みします。

(4) 拡張型の AI 識別子(ファンクションコード)について

決められている AI 識別子(ファンクションコード)は、何も指定しなくても、Barcode.php がコード体系から判断し自動的に挿入します。

ただし、任意で AI 識別子を挿入する場合は、{AI}を入力してください。

例：(01)商品識別コード + 任意の AI 0100012345678905{AI}10ABC123 ⇒ (01)00012345678905(10)ABC12

(5) 添字の描画

バーコードの下にコードの文字列自体を描画します(既定値)。プロパティの設定で描画をしないようにすることも可能です。

添字のフォントをプロパティで指定することも可能です。

(6) SVG出力 (Ver 4.0 新機能)

OutputSVG プロパティを true に設定すると、draw / draw_by_width メソッドの戻り値がSVG文字列になります。

3. アプリケーションプログラムからBarcode.phpの使用方法

3-1. クラス仕様

3-1-1. 概要

Barcode.php は、以下のそれぞれバーコードごとに独立したクラスで構成されております。

- Jan13
- Jan8
- UpcA
- UpcE
- ITF
- Matrix2of5
- NW7
- Code39
- Code128
- EAN128
- GS1_128 ... (旧名)EAN128を継承
- YubinCustomer
- QRCode
- DataMatrix
- Pdf417

一次元バーコードのクラスは基本的に同一名のプロパティやメソッドといったメンバを所有し、それらの機能も基本的に同一です。

そこで以降の各メンバの説明では、

- 一次元バーコードのクラスと
- YubinCustomer(郵便カスタマバーコードクラス)と、
- 二次元バーコード(QRCode / DataMatrix / Pdf417) のクラス

の3つに分けてご説明いたします。

3-1-2. 一次元バーコードクラスメンバ

3-1-2-1. コンストラクタ

初期処理を行う。

バーコードの種類別に以下のインターフェイスが存在します。

(1) JAN13() (2) JAN8() (3) UpcA() (4) UpcE() (5) ITF() (6) Matrix2of5() (7) NEC2of5() (8) NW7() (9) Code39() (10) Code93() (11) Code128() (12) EAN128() (13) GS1_128() ... (旧名)EAN128を継承

・引数 なし。

3-1-2-2. メソッド(関数)

(1) **function draw(\$code, \$minWidthDot, \$height)**

バーコードの描画を行います。

・引数

① \$code 描画を行うバーコードのコードを文字列で指定します。

少し特殊な GS1-128(UCC/EAN128)において、AI(アプリケーション識別子)挿入方法は2通りございます。

a) 可変長項目(データブロック)の後の AI には、FNC1 を挿入 ⇒これまで通り"{FNC1}"を付ける。例："{FNC1}21"のようにコードを指定

b) 固定長項目(データブロック)の後の AI には、固定長のため目印の FNC1 は不要 ⇒新しく追加した"{AI}"を付ける。例："{AI}21"のようにコードを指定

"{AI}"を指定して FNC1 を挿入しない場合も、カッコ()付コード文字は出力されます。例えば入力コードに" {AI}21"を指定した場合、添え字には(21)と出力されます。

例) (01)04512345670016(21)1 ↑(01)の前には FNC1 を挿入し(21)の前には挿入しない。コード指定方法 → " {FNC1}0104512345670016{AI}211"

② \$minWidthDot 横方向の最少描画ドット数指定します。(単位はドット)

③ \$height バーコードのバーの高さを指定します。(単位はドット)

・戻り値

OutputSVG = false の場合：バーコードの画像(image) OutputSVG = true の場合：SVG文字列

(2) function draw_by_width(\$code, \$width, \$height) 【Ver 4.0 新機能】

バーコード全体の幅を指定してバーコードの描画を行います。

・引数

- ① \$code 描画を行うバーコードのコードを文字列で指定します。
- ② \$width バーコード全体の幅を指定します。(単位はドット)
- ③ \$height バーコードのバーの高さを指定します。(単位はドット)

・戻り値

OutputSVG = false の場合 : バーコードの画像(image) OutputSVG = true の場合 : SVG文字列

(3) function draw_svg(\$code, \$minWidthDot, \$height) 【Ver 4.0 新機能】

SVG形式でバーコードの描画を行います。OutputSVG プロパティを true に設定して draw() を呼び出すのと同等です。

・引数

- ① \$code 描画を行うバーコードのコードを文字列で指定します。
- ② \$minWidthDot 横方向の最少描画ドット数を指定します。(単位はドット)
- ③ \$height バーコードのバーの高さを指定します。(単位はドット)

・戻り値

SVG文字列

(4) function draw_by_width_svg(\$code, \$width, \$height) 【Ver 4.0 新機能】

SVG形式で、バーコード全体の幅を指定してバーコードの描画を行います。

・引数

- ① \$code 描画を行うバーコードのコードを文字列で指定します。
- ② \$width バーコード全体の幅を指定します。(単位はドット)
- ③ \$height バーコードのバーの高さを指定します。(単位はドット)

・戻り値

SVG文字列

GS1-128(UCC/EAN128)コンビニバーコードメソッド

(5) **function drawConvenience(\$code, \$minWidthDot, \$height)**

コンビニバーコードの描画を行います。

・引数

- ① \$code 描画を行うバーコードのコードを文字列で指定します。
- ② \$minWidthDot 横方向の最少描画ドット数指定します。(単位はドット)
- ③ \$height バーコードのバーの高さを指定します。(単位はドット)

・戻り値

OutputSVG = false の場合 : バーコードの画像(image) OutputSVG = true の場合 : SVG文字列

3-1-2-3. プロパティ

(1) **\$TextWrite**

true : 添字の描画を行う。(既定値) false : 添字を描画しない。

(2) **\$FontName**

添字のフォント名。既定値は、"MS ゴシック"

(3) **\$FontSize**

添字のフォントサイズ。既定値は、10。

(4) **\$KurobarChousei**

黒バーの幅をドット単位で調整できます。プリンタによって PDF から印刷すると黒バーが太すぎて読み取れないようなケースに使用します。既定値は、0。

(使用例ソースコード)

```
// ↓↓↓黒バーを1ドット細くします。↓↓↓
// ただし、全体のバーコードのサイズを大きくしないと
// 1ドット細くすると細くなりすぎてしまいます。
$barcode->KuroBarCousei = -1;

// そこで、バーコードの横幅を4倍にします。
// 具体的には、バーコードを描画する最小ドット幅を
// 1ドットから4ドットへ変更します。
$barcode->minWidthDot = 4;

// できあがったバーコードの画像ファイルが大きくて
// その後、指定サイズで PDF にその画像を読み込むため問題ありません。
```

(5) **\$dispStartStopCode**

Code39/NW7 のみ使用可能なプロパティ true : スタート／ストップコードの描画を行う。false : スタート／ストップコードを描画しない。(既定値)

(6) **\$outputCode**

Code39/NW7 のみ使用可能なプロパティ 添字に出力するコード(スタート／ストップコード含む)。出力プロパティなので、指定する必要はない。

(7) \$OutputSVG 【Ver 4.0 新機能】

SVG出力モードを指定します。 true : SVG形式で出力する。draw/draw_by_width の戻り値がSVG文字列になる。 false : GD画像オブジェクトで出力する。(既定値)

(8) \$FitWidth 【Ver 4.0 新機能】

幅指定描画 (draw_by_width) 時の描画モードを指定します。 true : 指定幅に完全フィット。小数モジュール幅を使用。(既定値) false : 整数ドット単位で描画。指定幅以内に収まる最大サイズ。

(9) \$ForeColor 【Ver 4.0 新機能】

バーコードの前景色 (バーの色) をRGB配列で指定します。既定値は、array(0, 0, 0) (黒)

```
$barcode->ForeColor = array(0, 0, 128); // 紺色
```

(10) \$BackColor 【Ver 4.0 新機能】

バーコードの背景色をRGB配列で指定します。既定値は、array(255, 255, 255) (白)

```
$barcode->BackColor = array(255, 255, 200); // 薄い黄色
```

◇次のプロパティは、CODE128 / GS1_128(EAN128)のみで使用するプロパティです。

(11) \$CodeABC

CODE128 のコードセットには、次の種類があります。

CODE-A : フルアスキー CODE-B: 1桁の アルファ・ニューメリック CODE-C: 2桁の数字

本プロパティでは、次の指定が可能です。

CodeSet128 クラスの CONST 値	説明
----------------------------	----

const AUTO = 0 Barcode.net が自動でコードセットを組み合わせ最小幅のバーコードとする。

const CODE_A = 1 CODE-A

const CODE_B = 2 CODE-B

const CODE_C = 3 CODE-C

既定値 : CodeSet128.AUTO

3-1-3. GS1データバー(RSS)クラスメンバ

GS1データバーのクラスのインターフェイス(コンストラクタ・メソッド・プロパティ)は、基本的に一次元バーコードのものと変わりません。特別な機能のプロパティのみ追加されております。

従って、メソッドやプロパティの説明は、冗長にならない目的のためにも割愛させていただいている部分がございます。

3-1-3-1. コンストラクタ

初期処理を行う。

GS1データバーの種類別に以下のインターフェイスが存在します。

(1) Databar14() (2) DatabarLimited() (3) DatabarExpanded()

・引数 なし。

3-1-3-2. メソッド(関数)

(1) **function draw(\$code, \$minWidthDot, \$height)**

GS1データバーの描画を行います。

・引数

① \$code 描画を行うGS1データバーのコードを文字列で指定します。

《拡張型の AI 識別子(ファンクションコード)について》 決められている AI 識別子(ファンクションコード)は、何も指定しなくとも、コード体系から自動的に挿入されます。ただし、任意で AI 識別子を挿入する場合は、{AI}を入力してください。

例：(01)商品識別コード + 任意の AI 0100012345678905{AI}10ABC123 ⇒ (01)00012345678905(10)ABC12

② \$minWidthDot 横方向の最少描画ドット数指定します。(単位はドット)

③ \$height GS1データバーのバーの高さを指定します。(単位はドット)

・戻り値

OutputSVG = false の場合：GS1データバーの画像(image) OutputSVG = true の場合：SVG文字列

(2) **function draw_by_width(\$code, \$width, \$height) 【Ver 4.0 新機能】**

GS1データバー全体の幅を指定してバーコードの描画を行います。

・引数

① \$code 描画を行うGS1データバーのコードを文字列で指定します。

② \$width バーコード全体の幅を指定します。(単位はドット)

③ \$height バーコードの高さを指定します。(単位はドット)

・戻り値

OutputSVG = false の場合：GS1データバーの画像(image) OutputSVG = true の場合：SVG文字列

3-1-3-3. プロパティ

(1) \$TextWrite

添字の描画を行う。(既定値 : true)

(2) \$FontName

添字のフォント名。(既定値 : "MS ゴシック")

(3) \$FontSize

添字のフォントサイズ。(既定値 : 10)

(4) \$KurobarChousei

黒バーの幅をドット単位で調整できます。(既定値 : 0)

(5) \$outputCode

Code39/NW7 のみ使用可能なプロパティ 添字に出力するコード(スタート/ストップコード含む)。出力プロパティなので、指定する必要はない。

(6) \$OutputSVG 【Ver 4.0 新機能】

SVG出力モードを指定します。(既定値 : false)

(7) \$FitWidth 【Ver 4.0 新機能】

幅指定描画時の描画モードを指定します。(既定値 : true)

(8) \$ForeColor 【Ver 4.0 新機能】

バーコードの前景色をRGB配列で指定します。(既定値 : array(0, 0, 0))

(9) \$BackColor 【Ver 4.0 新機能】

バーコードの背景色をRGB配列で指定します。(既定値 : array(255, 255, 255))

◇次のプロパティは、GS1データバー(RSS)のみで使用するものです。

(10) \$SymbolType

GS1データバーのタイプです。限定型では使用しません。標準型・拡張型で次のように定義されています。

出力するデータバーのタイプを標準型・拡張型、各々で指定します。

GS1 DataBar RSS 14 のタイプ

GS1 DataBar RSS 14 のタイプ

class Databar14

const OMNIDIRECTIONAL = 0 標準型

const STACKED = 1 二層型

const STACKED_OMNIDIRECTIONAL = 2 標準二層型

GS1 DataBar RSS Expanded のタイプ

class DatabarExpanded

const UNSTACKED = 0 一層型

const STACKED = 1 多層型

3-1-4. 郵便カスタマバーコードクラスメンバ

3-1-4-1. コンストラクタ

初期処理を行う。

`YubinCustomer ()`

- ・引数なし。

3-1-4-2. メソッド(関数)

(1) `function draw($code, $size)`

郵便カスタマバーコードの描画を行います。

- ・引数

① `$code` 描画を行うバーコードのコードを文字列で指定します。

コードは・・・[郵便番号の数字部分7桁]+[郵便番号では不明部分の住所の英数字を「-」区切り]で、指定してください。

例) 〒116-0013 東京都荒川区西日暮里五丁目37番5号スタートアップオフィスA-207号室 コード指定方法
→「11600135-37-5-A-207」

② `$size` バーコードの大きさを指定します。

- ・戻り値

バーコードの画像(image)

3-1-5. QRコードクラスメンバ

3-1-5-1. コンストラクタ

初期処理を行う。

`QRCode()`

・引数なし。

3-1-5-2. メソッド(関数)

(1) `function draw_base($code)`

ベースとなる一番小さなQRコードを描画します。

・引数

① `$code` 描画を行うQRコードのコードを文字列で指定します。

・戻り値

バーコードの画像(image)

(2) `function draw_by_size($code, $size)`

サイズ(1,2,4,8,16)を指定して読み取り精度の高いQRコードを描画します。`draw_by_width` に比べ画像の拡縮を行わないため精度が高くなります。

・引数

① `$code` 描画を行うQRコードのコードを文字列で指定します。

② `$size 1,2,4,8,16` のいずれかを指定

・戻り値

`OutputSVG = false` の場合 : バーコードの画像(image) `OutputSVG = true` の場合 : SVG文字列

(3) **function draw_by_width(\$code, \$width)**

指定された幅に伸縮したQRコードを描画します。 draw_by_size に比べ、理論上、読み取り精度は低下する可能性があります。

・引数

① \$code 描画を行うQRコードのコードを文字列で指定します。

② \$width バーコードの幅 (単位 : ドット)

・戻り値

OutputSVG = false の場合 : バーコードの画像(image) OutputSVG = true の場合 : SVG文字列

(4) **function draw_by_width_svg(\$code, \$width) 【Ver 4.0 新機能】**

SVG形式で、指定された幅のQRコードを描画します。

・引数

① \$code 描画を行うQRコードのコードを文字列で指定します。

② \$width バーコードの幅 (単位 : ドット)

・戻り値

SVG文字列

3-1-5-3. プロパティ

(1) **\$version**

バージョン 1~40 を指定。既定値は、5。

(2) **\$error_level**

エラー訂正レベル : L・M・Q・H のいずれかを指定。既定値は、M。

(3) **\$OutputSVG 【Ver 4.0 新機能】**

SVG出力モードを指定します。(既定値 : false)

(4) **\$FitWidth 【Ver 4.0 新機能】**

幅指定描画時の描画モードを指定します。(既定値 : true)

(5) **\$ForeColor 【Ver 4.0 新機能】**

QRコードの前景色をRGB配列で指定します。(既定値 : array(0, 0, 0))

(6) \$BackColor 【Ver 4.0 新機能】

QRコードの背景色をRGB配列で指定します。(既定値 : array(255, 255, 255))

3-1-6. DataMatrixクラスメンバ

3-1-6-1. コンストラクタ

初期処理を行う。

DataMatrix()

- ・引数 なし。

3-1-6-2. メソッド(関数)

(1) **function draw_by_size(\$code, \$size)**

サイズ(1~)を指定して読み取り精度の高いDataMatrixを描画します。 draw_by_width に比べ画像の拡縮を行わないため精度が高くなります。

- ・引数

① \$code 描画を行うDataMatrixのコードを文字列で指定します。

② \$size サイズを指定(1~)

- ・戻り値

OutputSVG = false の場合 : バーコードの画像(image) OutputSVG = true の場合 : SVG文字列

(2) **function draw_by_width(\$code, \$width, \$height)**

指定された幅に伸縮したDataMatrixを描画します。 draw_by_size に比べ、理論上、読み取り精度は低下する可能性があります。

- ・引数

① \$code 描画を行うDataMatrixのコードを文字列で指定します。

② \$width バーコードの幅 (単位 : ドット)

③ \$height バーコードの高さ (単位 : ドット) 省略時は幅と同じ

- ・戻り値

OutputSVG = false の場合 : バーコードの画像(image) OutputSVG = true の場合 : SVG文字列

(3) `function draw_by_width_svg($code, $width, $height) 【Ver 4.0 新機能】`

SVG形式で、指定された幅のDataMatrixを描画します。

・引数

① \$code 描画を行うDataMatrixのコードを文字列で指定します。

② \$width バーコードの幅（単位：ドット）

③ \$height バーコードの高さ（単位：ドット）。省略時は幅と同じ。

・戻り値

SVG文字列

3-1-6-3. プロパティ**(1) `$CodeSize`**

シンボルコードサイズ 既定値：DxCodeSize::DxSzAuto(自動)

class DxCodeSize

DxSzRectAuto

DxSzAuto

DxSzShapeAuto

DxSz10x10 ~ DxSz144x144

DxSz12x36, DxSz16x36, DxSz16x48

(2) `$StringEncoding`

全角エンコーディング ("utf-8", "shift-jis", etc..) 既定値：utf-8

(3) `$OutputSVG 【Ver 4.0 新機能】`

SVG出力モードを指定します。(既定値：false)

(4) `$FitWidth 【Ver 4.0 新機能】`

幅指定描画時の描画モードを指定します。(既定値：true)

(5) `$ForeColor 【Ver 4.0 新機能】`

DataMatrixの前景色をRGB配列で指定します。(既定値：array(0, 0, 0))

(6) `$BackColor 【Ver 4.0 新機能】`

DataMatrixの背景色をRGB配列で指定します。(既定値 : array(255, 255, 255))

3-1-7. Pdf417クラスメンバ

3-1-7-1. コンストラクタ

初期処理を行う。

`Pdf417()`

- ・引数なし。

3-1-7-2. メソッド(関数)

(1) `function draw($code, $size)`

バーコードの描画を行います。バーコード全体の幅を指定するのではなく、バーを描画する最小単位のドット数を指定します。(1～)

- ・引数

① `$code` 描画を行うPDF417のコードを文字列で指定します。

② `$size ($minLinePitch)` 1～ サイズを指定

- ・戻り値

`OutputSVG = false` の場合：バーコードの画像(image) `OutputSVG = true` の場合：SVG文字列

(2) `function draw_by_width($code, $width, $height) 【Ver 4.0 新機能】`

PDF417全体の幅を指定して描画を行います。

- ・引数

① `$code` 描画を行うPDF417のコードを文字列で指定します。

② `$width` バーコード全体の幅（単位：ドット）

③ `$height` バーコードの高さ（単位：ドット）。省略時はアスペクト比から自動計算。

- ・戻り値

`OutputSVG = false` の場合：バーコードの画像(image) `OutputSVG = true` の場合：SVG文字列

(3) `function draw_svg($code, $size) 【Ver 4.0 新機能】`

SVG形式でPDF417の描画を行います。

・引数

- ① \$code 描画を行うPDF417のコードを文字列で指定します。
- ② \$size サイズを指定 (1~)

・戻り値

SVG文字列

(4) `function draw_by_width_svg($code, $width, $height) 【Ver 4.0 新機能】`

SVG形式で、PDF417全体の幅を指定して描画を行います。

・引数

- ① \$code 描画を行うPDF417のコードを文字列で指定します。
- ② \$width バーコード全体の幅 (単位：ドット)
- ③ \$height バーコードの高さ (単位：ドット)。省略時はアスペクト比から自動計算。

・戻り値

SVG文字列

3-1-7-3. プロパティ

(1) **\$SizeKind = SIZE_KIND_AUTO;**

データ列数・行数決定方法 既定値: SIZE_KIND_AUTO

```
/* データ列数・行数決定方法 */
/*! 自動サイズ(アスペクト比より決定) */
const SIZE_KIND_AUTO = 0;
/*! 指定列数(dataColumns)に従う */
const SIZE_KIND_COLUMNS = 1;
/*! 指定行数(dataRows)に従う */
const SIZE_KIND_ROWS = 2;
/*! 指定列数・行数両方に従う(長方形の最小値) */
const SIZE_KIND_COLUMNS_AND_ROWS = 3;
```

(2) **\$CodeRows = 3;**

出力行数指定。 SizeKind が出力行数指定の場合 = (自動サイズでない・列数指定でない場合)有効 既定値: 3

(3) **\$CodeCols = 3;**

出力データカラム数指定。 SizeKind が出力列数指定の場合 = (自動サイズでない・行数指定でない場合)有効 既定値: 3

(4) **\$ErrorLevel = 2;**

エラー訂正レベル: 0~8 既定値は: 2。

(5) **\$UseAutoErrorLevel = true;**

自動でエラー訂正レベルを決定(する・しない) 既定値は: する(true)

(6) **\$AspectRatio = 0.5;**

縦横アクセプト比: 0.5 / 1.0 / 2.0 等を指定。 既定値: 0.5

(7) **\$StringEncoding = "utf-8"**

全角エンコーディング ("utf-8", "shift-jis", etc..) 既定値 : utf-8

(8) **\$OutputSVG 【Ver 4.0 新機能】**

SVG出力モードを指定します。 (既定値 : false)

(9) **\$FitWidth 【Ver 4.0 新機能】**

幅指定描画時の描画モードを指定します。(既定値 : true)

(10) \$ForeColor 【Ver 4.0 新機能】

PDF417の前景色をRGB配列で指定します。(既定値 : array(0, 0, 0))

(11) \$BackColor 【Ver 4.0 新機能】

PDF417の背景色をRGB配列で指定します。(既定値 : array(255, 255, 255))

3-2. 使用例・サンプルプログラム

以下のサイトにサンプルプログラムをご用意してございます。

<http://www.pao.ac/barcode.php/sample/>

ここではその一つ、「ブラウザにバーコードを出力する使用例」を記載しておきます。

【draw_browser_barcode.php(バーコード描画部分)】

```
<?php
$kind = (string)$_GET['kind'];
$code = (string)$_GET['code'];
$x = (float)$_GET['x'];
$y = (float)$_GET['y'];
$minWidthDot = (float)$_GET['minWidthDot'];
$height = (float)$_GET['height'];
$dispStartStopCode = (bool)$_GET['dispStartStopCode'];
$rotateAngle = (int)$_GET['rotateAngle'];

require_once("Code39.php");
require_once("Code128.php");
require_once("GS1_128.php");
require_once("ITF.php");
require_once("Jan13.php");
require_once("Jan8.php");
require_once("UpcA.php");
require_once("UpcE.php");
require_once("Matrix2of5.php");
require_once("NEC2of5.php");
require_once("NW7.php");
require_once("QR.php");
require_once("YubinCustomer.php");

$barcode = null;
$qr = null;

switch ($kind)
{
    case "code39":
        $barcode = new Code39();
        $barcode->dispStartStopCode = $dispStartStopCode;
        break;
    case "code128":
        $barcode = new Code128();
        break;
    case "gs1-128":
        $barcode = new GS1_128();
        break;
    case "itf":
        $barcode = new ITF();
        break;
}
```

```
case "jan13":
    $barcode = new Jan13();
    break;
case "jan8":
    $barcode = new Jan8();
    break;
case "UPC-A":
    $barcode = new UpcA();
    break;
case "UPC-E":
    $barcode = new UpcE();
    break;
case "matrix2of5":
    $barcode = new Matrix2of5();
    break;
case "nec2of5":
    $barcode = new NEC2of5();
    break;
case "nw7":
    $barcode = new NW7();
    $barcode->dispStartStopCode = $dispStartStopCode;
    break;
default :
    print("Nothing Barcode !!\n");
    break;
}

// バーコード描画 (PNGファイルへ)
$img = $barcode->draw($code, $minWidthDot, $height);

// 回転
if($rotateAngle != 0) {
    $img = imagerotate($img, $rotateAngle, 0);
}

$fileName = "gazou/gazou.png";
ImagePNG($img, $fileName);

$size = GetImageSize($fileName);

print("<html>\n");
print("<head>\n");
print("<title>Barcode Test Output</title>\n");
print("</head>\n");
print("<body>\n");
print("<DIV id=\"Barcode\" style=\"position : absolute; width : ".
    ".$size[0]."px; height : ".$size[1]."px; z-index : 1; visibility : visible; left : ".
    " ".$x."px; top : ".$y."px;\">\n".
    "<IMG src=\"".$fileName."?".rand()."\" width=".$size[0].
    " height=".$size[1]." border=\"0\"></DIV>\n");
print("</body>\n");
print("</html>\n");
```


SVG出力の使用例 【Ver 4.0 新機能】

```
<?php
require_once("Jan13.php");

$barcode = new Jan13();

// SVG出力モードを有効化
$barcode->OutputSVG = true;

// SVG文字列を取得
$svg = $barcode->draw("4912345678904", 2, 50);

// HTTPヘッダーを設定してブラウザに出力
header("Content-Type: image/svg+xml");
echo $svg;
?>
```

幅指定描画の使用例 【Ver 4.0 新機能】

```
<?php
require_once("Code128.php");

$barcode = new Code128();

// 幅300ピクセル、高さ80ピクセルでバーコードを描画
$img = $barcode->draw_by_width("1234567890", 300, 80);

// PNG形式で保存
imagepng($img, "barcode.png");
imagedestroy($img);
?>
```

FitWidth プロパティの使用例 【Ver 4.0 新機能】

```
<?php
require_once("Jan13.php");

$barcode = new Jan13();

// ドットピッタリモード（精度優先）
$barcode->FitWidth = false;

// 幅300ピクセルで描画（実際の幅は300以下の整数ドット幅になる）
$img = $barcode->draw_by_width("4912345678904", 300, 50);

imagepng($img, "barcode_precise.png");
```

```
imagedestroy($img);  
?>
```

SVG + 幅指定の組み合わせ 【Ver 4.0 新機能】

```
<?php
require_once("QR.php");

$qr = new QRCode();
$qr->version = 5;
$qr->error_level = "M";

// SVG出力モードを有効化
$qr->OutputSVG = true;

// 幅200ピクセルでQRコードを描画
$svg = $qr->draw_by_width("https://www.pao.ac/", 200);

// ファイルに保存
file_put_contents("qrcode.svg", $svg);
?>
```

4. 使用条件等

4-1. 試用版と製品版

Barcode.php は、ご購入前に試用することができます。

試用版は、以下のURLよりダウンロードしてください。

<http://www.pao.ac/barcode.php/>

Docker版（推奨）

「`docker-compose up -d`」のみでインストール・起動を行えるセットをご用意しております。

- CentOS Stream 9 + PHP 8.3
- Ubuntu 24.04 LTS + PHP 8.5
- Debian 9 + PHP 5.6（旧版）

VMWare Player イメージ版

OSは、VMWare Player で動作するいくつかの Linux ディストリビューションをご用意しております。

- Ubuntu 24.04 LTS + PHP 8.5（最新版）
- その他旧版（Ubuntu 11.10, CentOS 5.5, Fedora 14, Debian 6）

サンプルプログラムは、各OS内の以下のディレクトリにご用意しております。

- `/var/www/html/sample`
- `/usr/local/apache2/htdocs/sample`

試用版は、ソースコードを見えないようにするために、弊社独自仕様で暗号化しております。

試用版の制限は、バーコードに「SAMPLE」という文字が入ります。

QRコードのお試し版の制限は、指定したコードの先頭に半角の「9」という文字が入ります。

製品版は、Barcode.php を購入(ユーザ登録)していただいたお客様に、お客様専用の製品版ダウンロード URL をメールでお送りいたします。

バージョンアップの際は、WEBサイトにてお知らせいたします。

バージョンアップをご希望のお客様は、大変お手数でございますが、メールにてアップグレード依頼をしてください。すぐにバージョンアップ版をメールにてお送りいたします。

4-2. 使用許諾

Barcode.php の使用について、Barcode.php の使用者(以下「利用者様」と称します)と有限会社パオ・アット・オフィス(以下「弊社」と称します)は、以下の各項目についての内容に同意するものとします。

1. Barcode.php の使用に関する使用許諾書

この使用許諾書は、利用者様がお使いのパソコンにおいて、Barcode.php を使用する場合に同意しなければならない契約書です。

2. 使用許諾書の同意

利用者様が Barcode.php を使用する時点で、本使用許諾書に同意されたものとします。同意されない場合は、Barcode.php を使用する事はできません。

3. ライセンス(使用権)の購入

利用者様が Barcode.php の製品版を使用して開発を行う場合には、1台の開発用コンピュータで Barcode.php を使用するにあたり、1ライセンスを購入する必要があります。

お客様環境等、開発コンピュータでないマシンで Barcode.php を使用する場合ライセンスは必要ありません。ランタイムライセンスフリーでございます。

4. 著作権

Barcode.php の著作権は、いかなる場合においても弊社に帰属いたします。

5. 免責

Barcode.php の使用によって、直接的、あるいは、間接的に生じた、いかなる損害に対しても、弊社は補償賠償の責任を負わないものとします。

6. 禁止事項

Barcode.php 及びその複製物を第三者に譲渡・貸与する事は出来ません。

Barcode.php を開発ツールとして再販/再配布することを禁止します。なお、モジュールとして組み込みを行い再販/再配布する場合は、開発ツールとしての再販/再配布には含まれませんので、OKです。

7. 保証の範囲

弊社は Barcode.php の仕様を予告無しに変更することがあります。その場合の利用者様に対する情報提供は、弊社 WebSite にて行う事とします。

8. 適用期間

本使用許諾条件は利用者様が Barcode.php を使用した日より有効です。利用者様が本使用許諾条件のいずれかの条項に違反した場合、又は、本許諾条件に同意出来ない場合は、利用者様は Barcode.php を一切使用出来ないものとします。

4-3. 代金支払い方法(ユーザ登録の方法)

Barcode.php の製品版をご利用頂ける場合は、ライセンスを購入して頂く必要があります。ライセンス形態及び代金支払方法は以下のとおりです。

必要なライセンス数の考え方

Barcode.php をインストールするパソコンの台数。

1ライセンス当たりの価格

20,000円(税込:22,000円) (ソースコード含む)

- バグフィックス等のバージョンアップは原則として無償とさせて頂きます。
- 大幅な機能追加等によるバージョンアップの場合には別ライセンスとさせていただく場合がございます。
- 本価格は Barcode.php の使用権に対するものです。カスタマイズや保守等の費用は一切含まれておりません。

お支払方法

(22,000円×ライセンス数)を下記口座へ銀行振込、または、郵便振替による送金をして下さい。

銀行名	支店名(コード)	口座番号	名義
三菱UFJ銀行	新宿支店(341)	普通 3831891	ユ) パオアットオフィス
PayPay銀行	すずめ支店(002)	普通 6461359	ユ) パオアットオフィス
郵便口座番号	名義		
00150-0-576845	有限会社 パオ・アット・オフィス		

※振込手数料は利用者様負担でお願い致します。

お支払いの通知と製品の送付

- 振り込みが完了した時点で、必ず弊社WebSiteの「Barcode.php 入金連絡フォーム」から入金のご連絡をお願いいたします。 <http://www.pao.ac/barcode.php/buy.html#form>
- 弊社では上記連絡を受けて入金確認を行い、利用者様専用の Barcode.php の製品ダウンロード URL を利用者様へ電子メールにてお送りさせて頂きます。
- 利用者様へは電子メール以外での製品の提供は原則として行いません。
- 製品の再送付は原則として行いません。製品のファイルは消去してしまわないように大切にお取り扱いください。
- お振り込み頂いても入金の連絡がない場合、こちらから振り込み人様の情報が分からぬいため、製品の送付が行えません。必ず入金連絡を行って頂くようお願いいたします。

見積書/納品書/請求書/領収証の発行、納品後のお支払いについて

見積書/納品書/請求書/領収証の発行は可能でございます。本製品納品後のお支払いも可能でございます。

<http://www.pao.ac/barcode.php/buy.html>

上記サイトでの手続きにより、弊社からの見積書/納品書/請求書/領収証の発行、及び、納品後のお客様からのお支払いを行えるようになっております。

Barcode.php 説明書 Version 4.0

Pao@Office

<https://www.pao.ac/barcode.php/>