

ブックカバー作成用グラフィックス関数解説

あらまし

C または Java 言語から(文庫本サイズの)ブックカバー用の SVG ファイルを作成するための特定用途専用グラフィックスライブラリです。[Processing](#) という言語とできるだけ同じ名前で描画関数を用意しています。しかし、いくつかの関数は簡略化されていますし、当然ながら、アニメーションやインタラクション関係の関数はありません。

座標系

初期状態では、紙の左上が原点で、x 軸は右向き、y 軸は下向きにのびています。(ふつう数学で使う座標系と y 軸の向きが逆です。) 長さの単位は mm(ミリメートル)です。

サイズは A4 紙横向きがデフォルトになっています。A4 紙のサイズは 横 297mm × 縦 210mm です。ブックカバーにしたとき、描いた図形が本の表面に現れるようにするには、だいたい (43, 31)–(253, 179) の座標の範囲 (横 210mm × 縦 148mm) に図形がおさまるようにして下さい。

#include (C言語) あるいは import (Java言語)

Java版では、

```
import static util.SVG Turtle.*;
```

としてください。

C版では、

```
#include "svg.h"
```

としてください。

関数一覧

基本的に C, Java とも関数名は共通です。ただし、C で unsigned int 型の引数は、Java ではただの int 型に、char* 型は String 型になります。逆に Java 版の boolean 型は C ではただの int 型になります。

C 版	Java 版	この文書
unsigned int	int	unsigned int
char*	String	char*
int	boolean	boolean

初期設定・その他

```
void setPageSize(double width, double height);
```

使用する場合は start() の前に呼び出します。紙のサイズを幅 width, 高さ height に設定します。A4 サイズの場合は void a4Portrait(), void a4Landscape() を使ってください。

```
void start(void);
```

描画の開始のときに呼び出します。

```
void finish(void);
```

描画の終了のときに呼び出します。

```
void rulers(void);
```

文庫本ブックカバーの折り目の目安となる細い線を描きます。

色・属性設定

初期状態は、線なし・塗潰し黒です。

```
void stroke(unsigned int color);
    線の色を設定します。色は 0xRRGGBB の形式で指定します。
void strokeWidth(double w);
    線の太さを設定します。
void strokeOpacity(double opacity);
    線の透明度を 0(完全透明)～1(完全不透明)の値で指定します。
void noStroke(void);
    線を描きません。
void fill(unsigned int color);
    塗潰しの色を設定します。色は 0xRRGGBB の形式で指定します。
void fillOpacity(double opacity);
    塗潰しの透明度を 0(完全透明)～1(完全不透明)の値で指定します。
void noFill(void);
    塗潰ししません。
```

基本図形

```
void line(double x1, double y1, double x2, double y2);
    (x1, y1) から (x2, y2) へ線分を描きます。
void rect(double x, double y, double w, double h);
    左上の頂点の座標が (x, y)、幅 w、高さ h の長方形を描きます。
void ellipse(double x, double y, double w, double h);
    中心の座標が (x, y)、幅 w、高さ h の楕円を描きます。
void triangle(double x1, double y1, double x2, double y2, double x3, double y3);
    3つの頂点の座標が (x1, y1), (x2, y2), (x3, y3) の三角形を描きます。
void textFont(char* fontName, double size);
    文字のフォントとサイズ(単位 mm)を設定します。(初期値は "MS-Mincho", 12mm です。)
    Windows上で SVGを閲覧する場合、fontName としては "serif", "sans-serif", "cursive", "MS 明
    朝", "MS ゴシック", "MS Pゴシック", "MS P明朝", (注: MとSとPは全角、空白は半角)
    "Arial", "Times New Roman", "Verdana", "Courier New", "Andale Mono", "Comic
    Sans MS", "Garamond", "Georgia", "Impact", "Tahoma", "Trebuchet MS" などが使えるは
    ずです。
void text(char* str, double x, double y, ...);(C 言語の場合)
void text(String str, double x, double y);(Java 言語の場合)
    文字列 str を座標 (x, y) に表示します。
```

折れ線・曲線・多角形

```
void beginShape(void);
    図形の定義を開始します。
void vertex(double x, double y);
    beginShape で定義を開始した図形に頂点 (x, y) を追加します。
void endShape(int close);(C 言語の場合)
    beginShape で定義を開始した図形の定義を終了します。closeが 0 以外のときは、図形を閉じ
    多角形を描きます。closeが 0 のときは、閉じずに折れ線になります。
void endShape(boolean close);(Java 言語の場合)
    beginShape で定義を開始した図形の定義を終了します。close が true のときは、図形を閉じ多
    角形を描きます。close が false のときは、閉じずに折れ線になります。
```

注意:

beginShape と endShape の間に vertex 以外の描画関係の関数を呼び出さないようにして下さい。

画像

```
void image(char* url, double x, double y, double w, double h);
```

インターネット上の画像を左上の頂点の座標が (x, y)、幅 w、高さ h の長方形内に描画します。

色関連のユーティリティ

```
int hsb1(double h, double s, double v);
```

fill 関数や stroke 関数の引数として与えるための色の値を h(色相), s(彩度), v(明度)から計算します。h(色相), s(彩度), v(明度)はそれぞれ 0 から 1 の範囲の数で指定します。

```
int hsb360(double h, double s, double v);
```

fill 関数や stroke 関数の引数として与えるための色の値を h(色相), s(彩度), v(明度)から計算します。h(色相)は 0 から 360 の範囲、s(彩度), v(明度)はそれぞれ 0 から 100 の範囲の数で指定します。

```
int hsl1(double h, double s, double l);
```

fill 関数や stroke 関数の引数として与えるための色の値を h(色相), s(彩度), l(輝度)から計算します。h(色相), s(彩度), l(輝度)はそれぞれ 0 から 1 の範囲の数で指定します。

```
int hsl360(double h, double s, double l);
```

fill 関数や stroke 関数の引数として与えるための色の値を h(色相), s(彩度), l(輝度)から計算します。h(色相)は 0 から 360 の範囲、s(彩度), l(輝度)はそれぞれ 0 から 100 の範囲の数で指定します。

```
int rgb1(double r, double g, double b);
```

fill 関数や stroke 関数の引数として与えるための色の値を 光の三原色 r(赤), g(緑), b(青)から計算します。r(赤), g(緑), b(青)はそれぞれ 0 から 1 の範囲の数で指定します。

```
int rgb255(double r, double g, double b);
```

fill 関数や stroke 関数の引数として与えるための色の値を 光の三原色 r(赤), g(緑), b(青)から計算します。r(赤), g(緑), b(青)はそれぞれ 0 から 255 の範囲の数で指定します。

```
int bw1(double v);
```

fill 関数や stroke 関数の引数として与えるための無彩色の値を 0(黒)から 1(白)の値で指定します。

```
int bw255(double v);
```

fill 関数や stroke 関数の引数として与えるための無彩色の値を 0(黒)から 255(白)の値で指定します。

```
int rotateH360(int color, double a);
```

色相を a 度変えた新しい色を計算します。

```
int rotateH(int color);
```

色相を“適当に”変えた新しい色を計算します。

乱数その他のユーティリティ

```
void randomSeed(unsigned int seed);
```

乱数の種を明示的に設定します。(デバッグ用に常に同一の乱数列を発生させたいときに 使用します。通常は呼ぶ必要はありません。)

```
void randomizeByTime(void);
```

乱数の種を現在時刻により、設定します。(start から randomizeByTime が呼ばれているので、通常は明示的に呼ぶ必要はありません。)

```
double randomInRange(double min, double max);
```

min から max の範囲の乱数を発生します。

```
double radians(double deg);
```

度をラジアンに変換します。

タートルグラフィックス関数

“亀”は最初ページの真ん中 (148.5, 105) に ペンを下げた状態で 0 度の向き(右)を向いています。

```
void forward(double len);
```

現在、向いている向きに len だけ移動します。

```
void backward(double len);
```

現在の向きと逆向きに線を描画せずに len だけ移動します。

```
void turn(double angle);
    右方向に angle 度回転します。(左方向に回転する時は負の数を渡します。)
void penUp(void);
    ペンを上げます。(この状態で移動しても線を描きません。)
void penDown(void);
    ペンを下げます。(この状態で移動すると線を描きます。)
void direction(double dir);
    現在の向きに関係なく、direction 度の方向を向きます。
void go(double x, double y);
    現在の位置に関係なく、(x, y) に移動します。この間、ペンの状態に関係なく、線は描きません。
void pushTurtle(void);
    現在の“亀”の位置と向きをバックアップします。
void popTurtle(void);
    “亀”の位置と向きを、以前 pushTurtle でバックアップしたものに戻します。
void say(char* str, ...); (C 言語の場合)
void say(String str); (Java 言語の場合)
    現在の“亀”の位置と向きに、文字列 str を描画します。このとき、“亀”は移動しません。
```

座標系変換関係

```
void translate(double x, double y);
    以降の描画に使用する座標系を (x, y) だけ平行移動します。
void rotate(double theta);
    以降の描画に使用する座標系を原点を中心に theta(単位ラジアン)回転します。
void scale(double sx, double sy);
    以降の描画に使用する座標系を縦方向に sx, 横方向に sy だけ拡大します。
void pushMatrix(void);
    現在、使用している座標系を保存します。
void popMatrix(void);
    最近 pushMatrix で保存した座標系を取り出します。
void resetMatrix(void);
    座標系を最初のものに戻します。
```

印刷の仕方

最新バージョンの Firefox で印刷します。

- 「ファイル」-「ページ設定」で「書式」の「印刷方向」を「横」、「余白」をすべて「0」、「ヘッダとフッタ」をすべて「なし」に設定して下さい。文庫版は A4(拡大/縮小 100%)で、印刷します。
- プリンタの方の設定でも、印刷の向きを「横」に設定して下さい。
- 「ファイル」-「印刷」で印刷して下さい。

なお、プリンタによってはどうしても左上に余計なマージンが取られ、印刷位置がずれることがあります。その場合は、“Adobe PDF” や “Microsoft XPS Document Writer” を使って PDF 形式や XPS 形式を経由すると、ピッタリの位置に印刷できるようです。

Koji Kagawa