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 (https://processing.org/) 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 版では、

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

としてください

```
/* 間違い */
#include <svg.h>
```

ではありませんので気を付けてください。

関数一覧

初期設定・その他

`void start(void);`
　　描画の開始のときに必ず呼び出します。
`void finish(void);`
　　描画の終了のときに必ず呼び出します。

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 strokeWeight(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。

`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。

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 text(char* str, double x, double y, ...);`

文字列 str を座標 (x, y) に表示します。

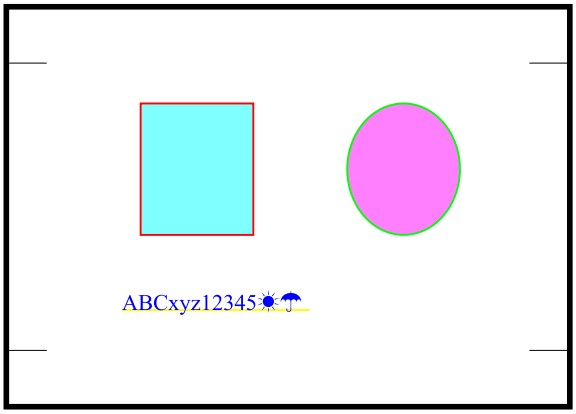
サンプルプログラム

C 版 C/FirstSample.c

```c
#include "svg.h"

int main(void){
    start();                        /* 最初に必要 */
    rulers();                       /* 四隅の線   */

    strokeWeight(1);
    stroke(hsb360(0, 100, 100)); /* 線の色 */
    fill(hsb360(180, 50, 100));  /* 塗りの色 */
    rect(70, 50, 60, 70);        /* 長方形 */

    stroke(hsb360(120, 100, 100));
    fill(hsb360(300, 50, 100));
    ellipse(210, 85, 60, 70);    /* 楕円 */

    stroke(hsb360(60, 100, 100));
    line(60, 160, 160, 160);     /* 直線 */

    noStroke();
    fill(hsb360(240, 100, 100));
```

```
21      textFont("Times New Roman", 12);
22      text("ABCxyz12345☀☂", 60, 160);
23      /* 文字列 */
24      finish();                       /* 最後に必要 */
25      return 0;
26  }
```

上記のプログラムが生成する図形（フルページで表示）



色関連のユーティリティ

`unsigned 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。

`unsigned 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。

`unsigned int rotateH360(unsigned int color, double a);`
色相を a 度変えた新しい色を計算します。

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 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。）

タートルグラフィックスのサンプルプログラム

C 版 C/TurtleSample.c

```c
#include "svg.h"

int main(void) {
    int i;

    start();                                    /* 最初に必要 */
    rulers();                                   /* 四隅の線   */

    direction(0);
    penUp();
    forward(60);
    penDown();
    for (i = 1; i <= 24; i++) {
        stroke(hsb360(i * 15, 100, 100)); /* 色 */
        forward(i * 3);                         /* 進む距離 */
        turn(90);                               /* 曲がる角度 */
    }

    center();
    direction(180);
    penUp();
    forward(60);
    penDown();
    for(i = 1; i <= 24; i++) {
        stroke(hsb360(i * 15, 100, 100)); /* 色 */
        forward(i * 3);                         /* 進む距離 */
        turn(90);                               /* 曲がる角度 */
    }

    finish();                                   /* 最後に必要 */
    return 0;
}

```

上記のプログラムが生成する図形（フルページで表示）