

## グラフィックス関数解説(抜粋)

### あらし

C 言語から SVG 画像ファイルを作成するための自家製グラフィックスライブラリーです。命令体系は標準的なもので Processing (<http://processing.org/>) という言語とできるだけ同じ名前でも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、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。ページのサイズはデフォルトで A4 サイズ (横 297mm × 縦 210mm) になっています。

### 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。

`void strokeWeight(double w);`

線の太さを設定します。

`void noStroke(void);`

線の描画なしに設定します。

`void fill(unsigned int color);`

塗潰しの色を設定します。

`void noFill(void);`

塗潰しなしに設定します。

`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", 12 です。Windows 上で SVG を閲覧する場合、フォントとしては "serif", "sans-serif", "monospace", "cursive", "fantasy", "MS 明朝", "MS ゴシック", "MS P明朝", "MS P ゴシック", (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が。

## 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 text(char str[], double x, double y);
```

文字列 str を座標 (x, y) に表示します。

## 色関連のユーティリティ

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

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);
```

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。

## サンプルプログラム

```
#include "svg.h" /* ライブラリー用のヘッダー */
/* #include <svg.h> ではないので注意する */

int main(void){
    start(); /* 最初に必要 */

    strokeWidth(1); /* 線の太さ */
    stroke(hsl360(0, 100, 50)); /* 線の色 */
    fill(hsl360(180, 50, 50)); /* 塗りの色 */
    rect(70, 50, 60, 70); /* 長方形 */

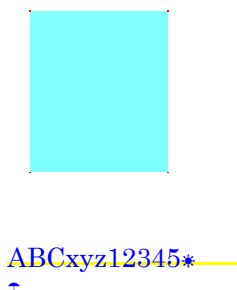
    stroke(hsl360(120, 100, 50));
    fill(hsl360(300, 50, 50));
    ellipse(210, 85, 60, 70); /* 楕円 */

    stroke(hsl360(60, 100, 50));
    line(60, 160, 160, 160); /* 直線 */

    noStroke();
    fill(hsl360(240, 100, 50));
    textFont("Times New Roman", 12);
    text("ABCxyz12345&#x2600;&#x2602;", 60, 160); /* 文字列 */

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

上記のプログラムが生成する図形



### タートルグラフィックス関数

“亀”（タートル）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);**

現在の向きに 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

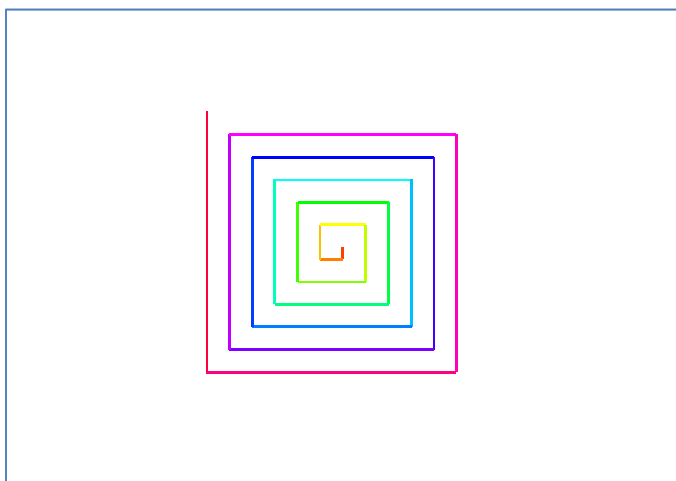
```
#include "svg.h"                /* ライブラリー用のヘッダー */

int main(void){
    int i;
    start();                      /* 最初に必要 */

    for (i = 0; i < 24; i++) {
        stroke(hsl360(i * 15, 100, 50));
        forward(i * 5);
        turn(90);
    }

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

上記のプログラムが生成する図形



### コンパイルと実行の仕方

Windows 上の Visual Studio で、*foo.c* というソースファイルをコンパイルする場合、

1. カレントディレクトリーに作成した *foo.c* とライブラリーのソースファイル *svg.c*、ライブラリーのヘッダーファイル *svg.h* を置きます
2. 次のコマンドで *svg.c* といっしょにコンパイルします。

```
cl foo.c svg.c
```

これで *foo.exe* というファイルができます

3. 次のコマンドで実行し、出力を *foo.svg* というファイルにリダイレクトします。  
*foo > foo.svg*
4. 生成された *foo.svg* を Firefox などのブラウザで開きます  
`start foo.svg`

### 印刷の仕方

Firefox などで印刷します。「ページ設定」で「書式とオプション」の「書式」の「印刷方向」を「横」、「余白とヘッダー/フッター」の「余白」をすべて「0」、「ヘッダーとフッター」をすべて「なし」に設定して下さい。A4（拡大/縮小 100%）で、印刷します。