

Processing クイックリファレンス

変数型・クラスの宣言

<code>int k;</code>	整数型変数 k を宣言する
<code>float x;</code>	実数型変数 x を宣言する
<code>color c;</code>	color 型変数 c を宣言する
<code>PImage f;</code>	画像オブジェクト f を宣言する

配列の宣言

<code>int[] k = new int[n];</code>	要素数 n の整数型配列 k を宣言する
<code>float[] k = new float[n];</code>	要素数 n の実数型配列 k を宣言する

型変換

<code>int(a)</code>	数値 a を整数型に変換する
<code>float(a)</code>	数値 a を実数型に変換する
<code>color(R, G, B)</code>	数値 R, G, B から color 型の値をつくる

描画関数 R, G, B は色の赤成分、緑成分、青成分を表す

<code>size(w, h)</code>	実行画面を横 w 画素、縦 h 画素にする
<code>stroke(R, G, B)</code>	図形の輪郭の色を指定する
<code>fill(R, G, B)</code>	図形の塗りつぶしの色を指定する
<code>line(x1, y1, x2, y2)</code>	座標 $(x1, y1)$ 、 $(x2, y2)$ の間に直線を描く
<code>rect(x, y, w, h)</code>	座標 (x, y) を左上頂点とする横 w 画素、縦 h 画素の矩形を描く
<code>println(data, ...)</code>	コンソールに文字や式の値を表示する

画像処理関数

<code>createImage(w, h, RGB)</code>	横 w 画素、縦 h 画素のカラー画像を新しく作る
<code>loadImage("file")</code>	画像ファイル $file$ から画像を読み込む
<code>f.save("file")</code>	画像 f をファイル $file$ に保存する
<code>f.filter(GRAY)</code>	画像 f をグレースケール画像に変換する
<code>f.filter(THRESHOLD, t)</code>	画像 f を閾値 t ($0 \leq t \leq 1$) で 2 値化する
<code>image(f, x, y)</code>	画像 f を実行画面の座標 (x, y) に表示する
<code>f.get(x, y)</code>	画像 f の座標 (x, y) の色 (color 型) を取得する
<code>f.set(x, y, c)</code>	画像 f の座標 (x, y) の色を c にする
<code>red(c)</code>	color 型変数 c から赤成分の画素値を取り出す (実数)
<code>green(c)</code>	color 型変数 c から緑成分の画素値を取り出す (実数)
<code>blue(c)</code>	color 型変数 c から青成分の画素値を取り出す (実数)

数学関数

<code>sin(x) cos(x) tan(x)</code>	各三角関数の値 x の単位はラジアン
<code>atan2(x, y)</code>	ベクトル (x, y) と x 軸の間の角度 (ラジアン)
<code>radians(x)</code>	角度 x の単位を度からラジアンに変換した値
<code>pow(x, y)</code>	x の y 乗 (x^y) の値
<code>exp(x)</code>	e の x 乗 (e^x) の値
<code>abs(x)</code>	x の絶対値
<code>sqrt(x)</code>	x の平方根
<code>max(x, y, z) min(x, y, z)</code>	x, y, z の最大値または最小値 引数は x, y の 2 つでもよい