

プログラミング

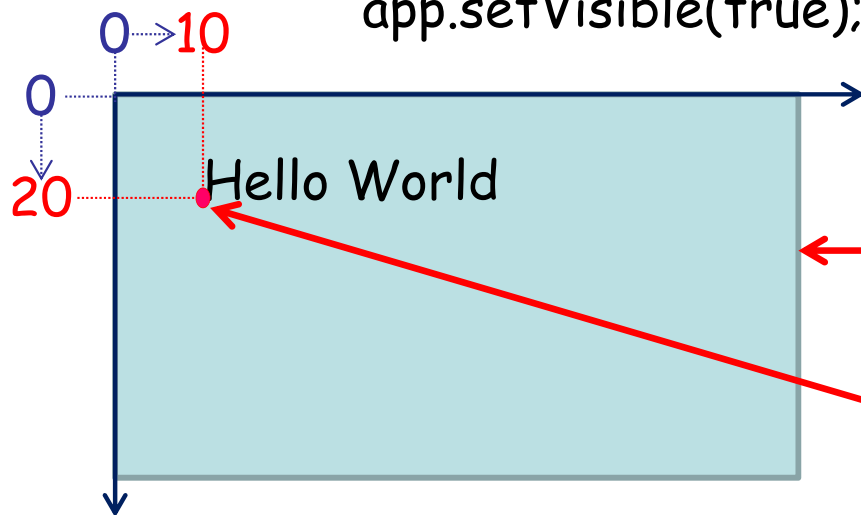
Javaプログラミング

堀田 敬介

図形描画

```
import java.awt.*;  
import javax.swing.*;
```

```
public class クラス名 extends JPanel { // Jpanelクラス継承  
    public void paintComponent(Graphics g) { // 描画処理メソッド  
        g.drawString("Hello World", 10, 20);  
    }  
    public static void main(String[] args) {  
        JFrame app = new JFrame(); // JFrameのインスタンス生成し変数appへ  
        app.add(new クラス名()); // 窓[app]内に[クラス]のインスタンスをはめ込む  
        app.setTitle("タイトル"); // 窓[app]に標題「タイトル」をつける  
        app.setSize(400, 400); // 窓[app]サイズ設定:幅400,高さ400  
        app setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        app.setVisible(true); // ↑窓[app]閉じるボタンで実行停止  
        // 窓[app]を目に見える(Visible)ようにする  
    }  
}
```



文字の書き出し左下位置を
座標(10,20)に設定するということ

色の設定法

```
import java.awt.*;
import javax.swing.*;
```

```
public class ... extends JPanel {
    Color col;           // カラークラス
    int rr, gg, bb;
```

```
public void paintComponent(Graphics g) {
    g.setColor(Color.red); // 色を設定するメソッド(準備色指定)
```

```
    col = new Color(0x00FF99); // 16進数で光3原色をつくる
    g.setColor(col);           // 作った色を設定
```

```
    rr = 255; gg = 0; bb = 125;
    col = new Color(rr, gg, bb); // 10進数で光3原色をつくる
    g.setColor(col);           // 作った色を設定
```

```
}
```

```
}
```

光の強さ
【16進数】

弱	00	10	...	F0
	01	11	...	F1
	:	:	...	:
	0F	1F	...	FF強

光の強さ
【10進数】

弱	0	16	...	240
	1	17	...	241
	:	:	...	:
	15	31	...	255強

色	指定法	色	指定法
白	<i>Color.white</i>	赤	<i>Color.red</i>
薄灰	<i>Color.lightGray</i>	青	<i>Color.blue</i>
灰	<i>Color.Gray</i>	緑	<i>Color.green</i>
濃灰	<i>Color.darkGray</i>	淡紅	<i>Color.pink</i>
黒	<i>Color.black</i>	橙	<i>Color.orange</i>
		シアン	<i>Color.cyan</i>
		マゼンタ	<i>Color.magenta</i>
		イエロー	<i>Color.yellow</i>

準備色8色 + 濃淡5色



フォントの設定法

```
import java.awt.*;  
import javax.swing.*;
```

```
public class ... extends JPanel {  
    Font ft;           // フォントクラス  
  
    public void paintComponent(Graphics g) {  
        ft = new Font("Elephant", Font.PLAIN, 24); // フォントをつくる  
        g.setFont(ft); // フォント設定メソッド  
    }  
}
```

サイズ



// フォントをつくる

// フォント設定メソッド

指定できるフォントの種類(例)

指定できる書体

フォント(英字)	フォント(日本語)
Rockwell	MS ゴシック
Broadway	MS 明朝
Times New Roman	HGP行書体
Georgia	HG教科書体
Tekton Pro	null (指定せず)

書体	意味
Font.PLAIN	普通
Font.BOLD	太字
Font.ITALIC	斜体
Font.BOLD Font.ITALIC	太字斜体

基本図形描画

```
import java.awt.*;
import javax.swing.*;

public class ... extends JPanel {
    public void paintComponent(Graphics g) {
        g.drawLine(10, 10, 100, 30);           // 線分

        g.drawRect(10, 10, 100, 50);          // 矩形
        g.fillRect(10, 10, 100, 50);          // 塗り潰し矩形

        g.drawOval(10, 10, 70, 50);           // 楕円
        g.fillOval(10, 10, 70, 50);          // 塗り潰し楕円

        g.drawRoundRect(10, 10, 100, 50, 5, 5); // 角丸矩形
        g.fillRoundRect(10, 10, 100, 50, 5, 5); // 塗り潰し角丸矩形

        int x[] = {10, 60, 70}, y[] = {30, 30, 60}, pt = 3;
        g.drawPolygon(x, y, pt);              // 多角形
        g.fillPolygon(x, y, pt);             // 塗り潰し多角形
    }
}
```

スレッド

```
import java.awt.*;  
import javax.swing.*;
```

```
public class クラス名 extends JPanel implements Runnable {  
    Thread thd = null;
```

```
    public クラス名() {          // コンストラクタ  
        thd = new Thread(this); // 自クラスでスレッド利用  
        thd.start();           // スレッド開始  
    }
```

```
    public void run() {          // スレッド開始時に呼び出されるメソッド  
        ...                     // 実行する命令群  
        repaint();              // 再描画 (paintComponent 実行)  
    }
```

```
    public void paintComponent(Graphics g) { ... }  
    public static void main(String[] args) { ... }
```

```
}
```

スレッド (一時停止)

```
import java.awt.*;
import javax.swing.*;

public class クラス名 extends JPanel implements Runnable {
    Thread thd = null;
    public クラス名() { // コンストラクタ
        thd = new Thread(this); // 自クラスでスレッド利用
        thd.start(); // スレッド開始
    }
    public void run() { // スレッド開始時に呼び出されるメソッド
        ... // 実行する命令群
        repaint(); // 再描画
        try {
            thd.sleep(500); // スレッド一時停止(例:500ミリ秒停止)
        } catch (InterruptedException e) {
            e.printStackTrace(); // スタックトレース
        }
    }
    public void paintComponent(Graphics g) { ... }
    public static void main(String[] args) { ... }
}
```