

アプレット

```
import java.applet.*;  
import java.awt.*;
```

```
public class ... extends Applet {
```

```
    public void paint(Graphics g) { // アプレット 描画  
        g.drawString("Hello World, 10, 20");  
    }
```

```
}
```

アプレット:色

```
import java.applet.*;
import java.awt.*;

public class ... extends Applet {
    Color col;          // カラークラス
    int red, grn, blu;

    public void paint(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

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

アプレット:色

準備されている色は以下の8色 + 白黒5色

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

アプレット: フォント

```
import java.applet.*;  
import java.awt.*;
```

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

指定できるフォントの例

フォント	指定法
	Font.PLAIN
	Font.

アプレット: 基本図形描画

```
import java.applet.*;  
import java.awt.*;
```

```
public class ... extends Applet {  
    public void paint(Graphics g) {  
        g.drawLine(10, 100);  
  
        g.drawRect(10, 10, 100, 50);  
        g.fillRect(10, 10, 100, 50);  
  
        g.drawOval(10, 10, 70, 50);  
        g.fillOval(10, 10, 70, 50);  
  
        g.drawRoundRect(10, 10, 100, 50, 5, 5);  
        g.fillRoundRect(10, 10, 100, 50, 5, 5);  
  
        int x[] = {10, 60, 70}, y[] = {30, 30, 60}, pt = 3;  
        g.drawPolygon(x, y, pt);  
        g.fillPolygon(x, y, pt);  
    }  
}
```

グラフィックス

```
import java.applet.*;
import java.awt.*;

public class ... extends Applet {
    Graphics g;          // グラフィックスクラス

    public void paint(Graphics g) {
        test_disp();
    }

    public void test_disp() { //paint()以外でグラフィックスを作りまとめて描画
        g = getGraphics();  // グラフィックス取得メソッド
        g.drawLine(10, 10, 50, 70);
        g.fillOval(60, 30, 15, 20);

        repaint();        // アプレット強制再描画
    }
}
```

スレッド

```
import java.applet.*;
import java.awt.*;

public class ... extends Applet implements Runnable {
    Thread thd = null;

    public void init() {           // 初期化処理
        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

        repaint();               // アプレット強制再描画
    }
}
```

スレッド (一時停止)

```
import java.applet.*;
import java.awt.*;

public class ... extends Applet implements Runnable {
    Thread thd = null;

    public void init() { // 初期化处理
        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

        repaint(); // アプレット強制再描画

        try {
            thd.sleep(500); // スレッド一時停止(例:500ミリ秒停止)
        } catch (InterruptedException e) { }
    }
}
```


イベント処理: マウス使用

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;
public class ... extends Applet implements MouseListener {
    public void init() {          // 初期化处理
        addMouseListener(this); // MouseEventを自クラスで受け取る宣言
    }
    // マウスがアプレット領域内へ入った時の処理 (使わなくても必要)
    public void mouseEntered(MouseEvent e) { }
    // マウスがアプレット領域外へ出た時の処理 (使わなくても必要)
    public void mouseExited(MouseEvent e) { }
    // マウス・ボタン押下時処理 (使わなくても必要)
    public void mousePressed(MouseEvent e) { }
    // マウス・ボタン押下状態→放した時の処理 (使わなくても必要)
    public void mouseReleased(MouseEvent e) { }
    // マウスクリック時の処理 (使わなくても必要)
    public void mouseClicked(MouseEvent e) {
        Point pt = e.getPoint(); // マウス位置取得メソッド
        x = pt.x; y = pt.y;
    }
}
```

イベント処理: キーボード入力

```
public class ... extends Applet implements KeyListener {
    char key;
    int key2;

    public void init() {           // 初期化処理
        addKeyListener(this);     // キー入力Eventを自クラス受取宣言
    }

    public void paint(Graphics g) { // 描画処理
        requestFocusInWindow(); // アプレットフォーカスに必要
    }

    public void keyTyped(KeyEvent e) { // キーが押された時の処理
        key = e.getKeyChar();         // 文字取得メソッド
    }

    public void keyPressed(KeyEvent e) { // キーが押された時の処理
        key2 = e.getKeyCode();        // 文字code取得メソッド
    }

    public void keyReleased(KeyEvent e) { } // キーが離された時の処理
}
}
```

ボタン, アクションイベント

```
import java.applet.*;
import java.awt.*;
import java.awt.event.*;

public class ... extends Applet implements ActionListener{
    Button btn;                // ボタン : 宣言

    public void init() { // 初期化处理
        btn = new Button("...");
        add(btn);
        btn.addActionListener(this); // ActionEvent自クラス受取宣言
    }

    public void actionPerformed(ActionEvent e) { // ActionEvent実行処理
        if (e.getSource() == btn) { // ボタンが押されたら...
        }
    }
}
}
```

アプレット領域のサイズ取得

```
import java.applet.*;
import java.awt.*;

public class ... extends Applet {
    Dimension siz;           // サイズ : 宣言
    int app_wid, app_hei;

    public void init() {    // 初期化处理
        siz = getSize();    // アプレット画面の大きさ取得
        app_wid = siz.width; // アプレット領域の幅取得
        app_hei = siz.height; // アプレット領域の高さ取得
    }
}
```