



Power Java

제26장 애플릿



© 2009 인피니티박스 All rights reserved



이번 장에서 학습할 내용

- 애플릿 소개
- 애플릿 작성 및 소개
- 애플릿의 생명주기
- 애플릿에서의 그래픽 컴포넌트의 소개
- Applet API의 이용

웹
브라우저상에서 실행되는
작은
프로그램인
애플릿에
대하여
학습합니다.

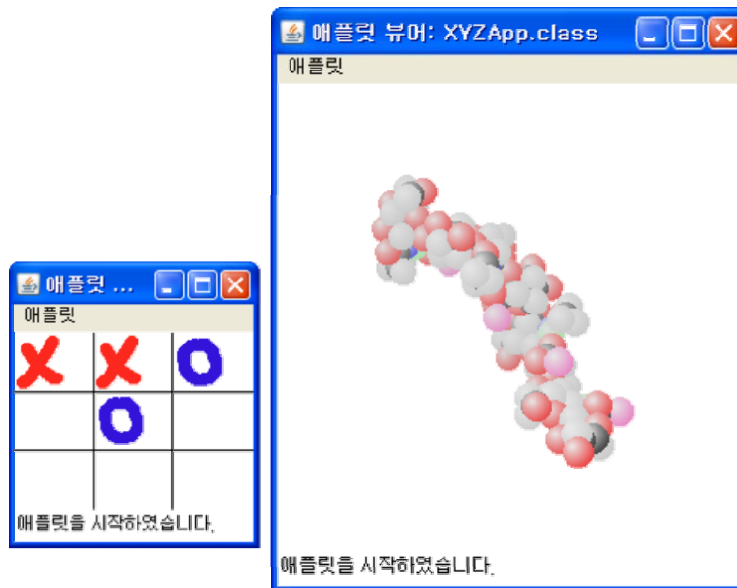


© 2009 인피니티박스 All rights reserved



애플릿이란?

- 애플릿은 웹 페이지 같은 HTML 문서 안에 내장되어 실행되는 자바 프로그램이다.



.7림 26-1 샘플 애플릿

© 2009 인피니티박스 All rights reserved



애플릿을 실행시키는 두가지 방법

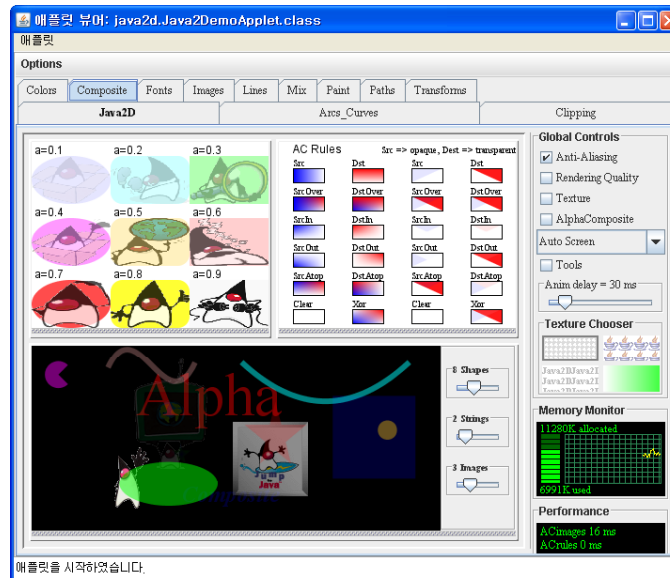
1. 웹 브라우저를 이용하는 방법
2. Appletviewer를 이용하는 방법
 - (예) C>appletviewer example1.html

© 2009 인피니티박스 All rights reserved



애플릿에서 할 수 있는 것

- 그래픽을 포함한 거의 모든 자바 라이브러리를 사용할 수 있다.
- (예)



© 2009 인피니티박스 All rights reserved



애플릿과 애플리케이션의 비교

- 애플릿
 - main() 메소드를 가지지 않음
 - 웹을 통하여 쉽게 실행
 - 현재 동작되는 컴퓨터를 변경할 수 없음
 - 보안 측면에서 우수
- 애플리케이션
 - main() 메소드를 가지고 여기서부터 모든 작업이 시작
 - 자바 웹 스타트(Java Web Start) 기능을 이용하면 쉽게 웹을 통하여 다운로드되어서 실행

© 2009 인피니티박스 All rights reserved



전형적인 애플릿의 구조

```
public class MyApplet extends JApplet {
    public void init() { // 애플릿 초기화 }

    public void start() { // 애플릿의 실행 시작 }

    public void stop() { // 애플릿의 실행 정지 }

    public void destroy() { // 애플릿의 종료 }

    public void paint(Graphics g) { // 그림을 그리는 코드 }
    ...
}
```

애플릿 위에
그림을 그린다.

© 2009 인피니티북스 All rights reserved



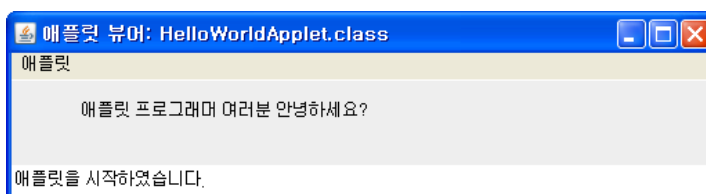
애플릿 #1

```
import java.awt.Graphics;
import javax.swing.JApplet;

public class MyApplet extends JApplet {
    public void paint(Graphics g) {
        super.paint(g);
        g.drawString("애플릿 프로그래머 여러분 안녕하세요?", 50, 25);
    }
}
```

JApplet을
상속받는다.

Paint() 메소드를
재정의한다.



© 2009 인피니티북스 All rights reserved



애플릿을 실행시키려면

- Test.html 파일을 작성한다.

```
<html>
  <body>
    <applet code="MyApplet.class" width=400 height=100>
    </applet>
  </body>
</html>
```

- Appletviewer로 실행한다.



```
C> appletviewer Test.html
```



애플릿이 들어있는 위치 지정

- 절대 경로 형식

```
<applet codebase="http://test.sch.ac.kr/pub/java"
        code="MyApplet.class" width=200 height=80>
</applet>
```

- 상대 경로 형식

```
<applet codebase="pub/java" code="MyApplet.class" width=200
        height=80>
</applet>
```



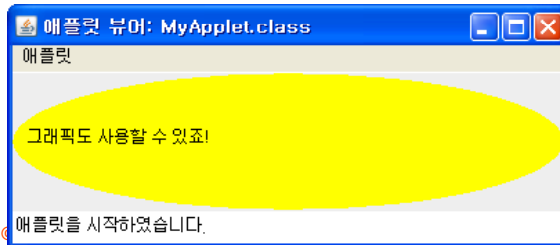
애플릿 #2

```
import java.awt.Color;
import java.awt.Graphics;
import javax.swing.JApplet;

public class MyApplet1 extends JApplet {
    public void paint(Graphics g) {
        super.paint(g);
        g.setColor(Color.YELLOW);
        g.fillOval(0, 0, getSize().width - 1, getSize().height - 1);
        g.setColor(Color.BLACK);
        g.drawString("그래픽도 사용할 수 있죠!", 10, getSize().height / 2);
    }
}
```

Paint() 메소드를 재정의한다.

그래픽 메소드 사용

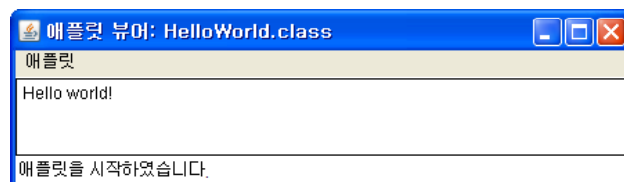


애플릿 #3

```
import javax.swing.JApplet;
import java.awt.Graphics;

public class HelloWorld extends JApplet {
    public void paint(Graphics g) {
        g.drawRect(0, 0, getSize().width - 1, getSize().height - 1);
        g.drawString("Hello world!", 5, 15);
    }
}
```

애플릿의 크기





애플릿의 생명 주기

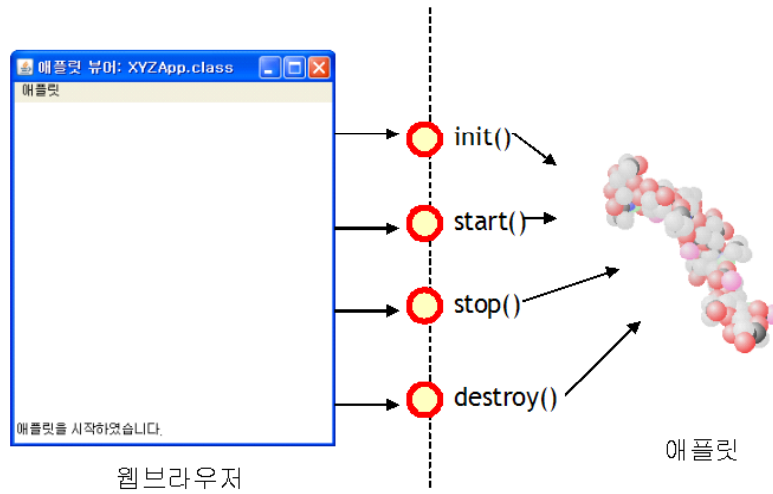


그림 26-3 웹브라우저와 애플릿 인터페이스

© 2009 인피니티박스 All rights reserved



애플릿과 웹브라우저 인터페이스

- **init()**
 - 애플릿을 초기화하는 메소드
 - 애플릿이 들어있는 페이지를 열 때 자동으로 호출
- **start()**
 - 사용자가 애플릿이 포함된 페이지를 방문할 때마다 호출
- **stop()**
 - 사용자가 애플릿이 포함된 페이지를 떠날 때 호출된다.
- **destroy()**
 - 브라우저가 닫힐 때만 호출된다.

© 2009 인피니티박스 All rights reserved



애플릿 예제

```
import java.awt.Graphics;
import javax.swing.JApplet;

public class MyApplet2 extends JApplet {
    StringBuffer output;

    public void init() {
        output = new StringBuffer();
        output.append("init() ");
        repaint();
    }

    public void start() {
        output.append("start() ");
        repaint();
    }
}
```

© 2009 인피니티박스 All rights reserved



애플릿 #2

```
public void stop() {
    output.append("stop() ");
    repaint();
}

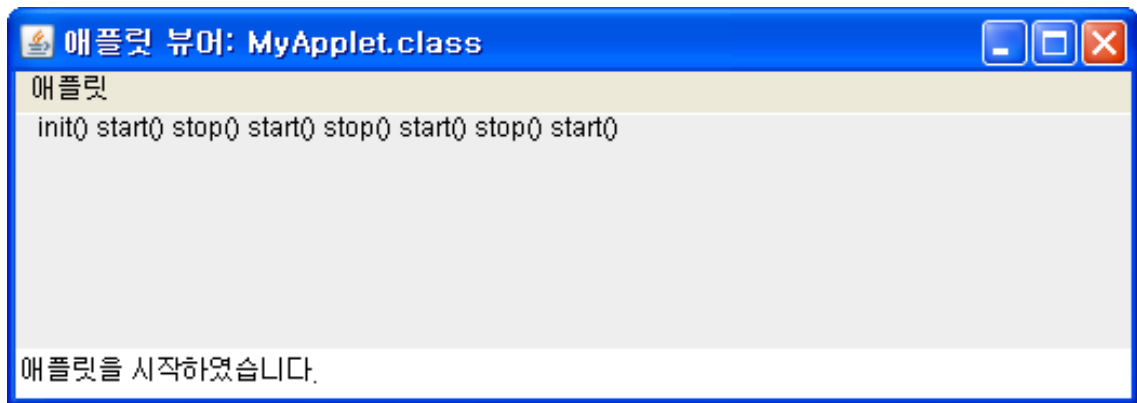
public void destroy() {
    output.append("destroy() ");
    repaint();
}

public void paint(Graphics g) {
    super.paint(g);
    g.drawString(output.toString(), 10, 10);
}
}
```

© 2009 인피니티박스 All rights reserved



실행 결과

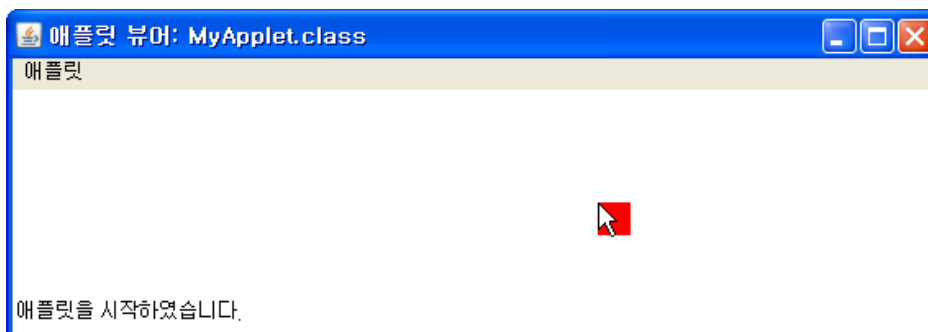


© 2009 인피니티박스 All rights reserved



애플릿 위에서 GUI 사용

- 모든 GUI 컴포넌트를 사용할 수 있다.
- (예) 다음과 같이 마우스를 클릭하면 사각형을 그리는 애플릿을 작성하여 보자.



© 2009 인피니티박스 All rights reserved



애플릿 예제

```
import java.awt.event.*;
import java.applet.Applet;
import java.awt.Color;
import java.awt.Graphics;

public class MyApplet3 extends Applet implements MouseListener {
    int x, y;

    public void init() {
        addMouseListener(this);           // 마우스 리스너를 등록한다.
    }

    public void paint(Graphics g) {
        super.paint(g);
        g.setColor(Color.RED);
        g.fillRect(x, y, 20, 20);         // 마우스의 현재 위치에 사각형을 그린다.
    }
}
```

© 2009 인피니티박스 All rights reserved



애플릿 예제

```
public void mouseEntered(MouseEvent event) { }

public void mouseExited(MouseEvent event) { }

public void mousePressed(MouseEvent event) { }

public void mouseReleased(MouseEvent event) { }

public void mouseClicked(MouseEvent event) {
    x = event.getX();           // 마우스가 클릭되면 현재 위치를 얻는다.
    y = event.getY();
    repaint();                 // 다시 화면을 그린다.
}
}
```

paint()를 직접
호출하는 것이 아니라
repaint()를 호출한다.

© 2009 인피니티박스 All rights reserved



애플릿 예제

```
import java.awt.event.*;
import java.applet.Applet;
import java.awt.*;
import javax.swing.*;

public class MyApplet4 extends JApplet implements ActionListener {
    JTextField field;
    JTextArea area;

    public void init() {
        field = new JTextField();
        area = new JTextArea();
        setLayout(new BorderLayout());
        add(field, BorderLayout.SOUTH);
        add(area, BorderLayout.CENTER);
        field.addActionListener(this);
    }
}
```

배치 관리자를
BorderLayout 으로
변경

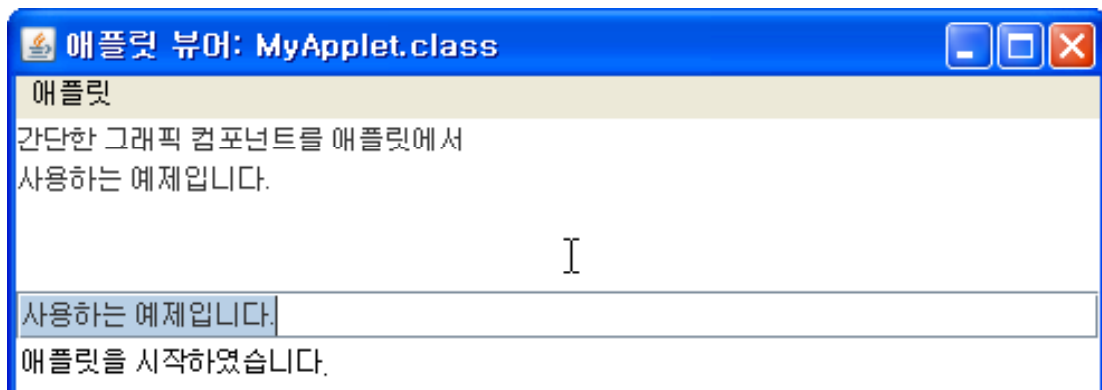
© 2009 인피니티박스 All rights reserved



애플릿 예제

```
public void actionPerformed(ActionEvent arg0) {
    area.append(field.getText() + "\n");
    field.selectAll();
    area.setCaretPosition(area.getDocument().getLength());
    repaint();
}
}
```

이벤트 처리기 작성.



© 2009 인피니티박스 All rights reserved



애플릿 매개 변수

- 애플릿과 웹브라우저는 매개 변수를 통하여 값들을 주고 받을 수 있다.

HTML
파일

```
<APPLET CODE="test.class" WIDTH=200 HEIGHT=200>
<PARAM NAME="imgDir" VALUE="java/images">
<PARAM NAME="bColor" VALUE="0xff00c0">
<PARAM NAME="COUNT" VALUE=20>
...
</APPLET>
```

애플릿

```
int count = 0;
...
String countString = getParameter("COUNT");
if (wString != null) {
    try {
        count = Integer.parseInt(countString);
    } catch (NumberFormatException e) {
        count = 10;    // 디폴트값을 사용
    }
}
```



애플릿이 할 수 없는 것

- 실행받지 못한 애플릿이 할 수 없는 기능은 다음과 같다.
 - 라이브러리를 적재하거나 네이티브 메소드를 정의할 수 없다.
 - 수행되고 있는 호스트 컴퓨터의 파일을 읽고 쓸 수 없다.
 - 네트워크를 통한 연결을 만들 수 없다.
 - 호스트 컴퓨터상의 프로그램을 실행시킬 수 없다.
 - 호스트 컴퓨터의 시스템 특성을 읽을 수 없다.



Q & A

