

## 제 15 장 애플릿, 멀티미디어

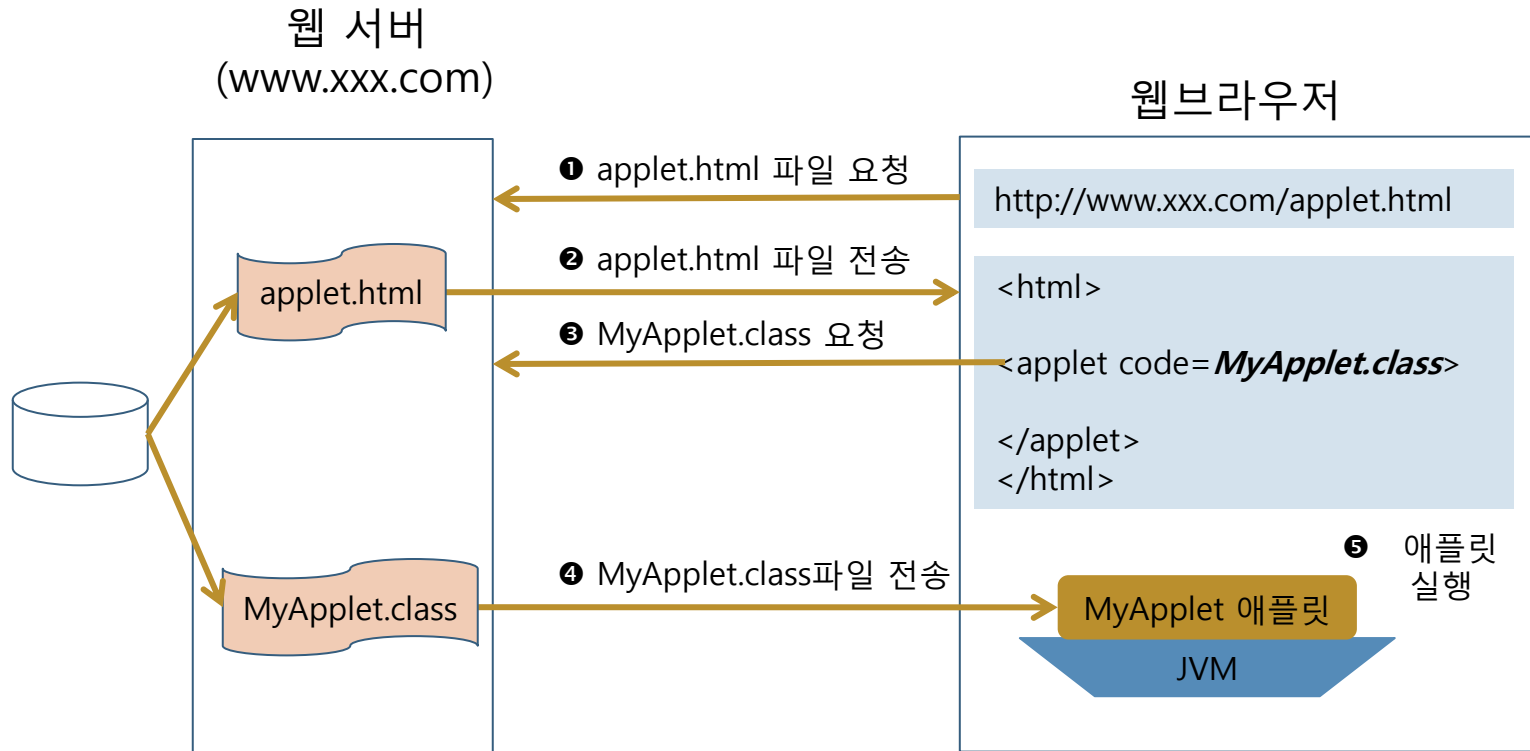
# 자바 애플릿

2

- 애플릿 프로그램의 실행 환경
  - ▣ 웹 브라우저 내에서 실행되는 자바 응용 프로그램
  - ▣ HTML 페이지에 내장되어 실행됨
  - ▣ `main()` 메소드 없음
    - 웹 브라우저에 의해 실행되고 소멸되는 구조
    - 웹 브라우저에 의해 애플릿 코드에 작성된 메소드를 호출하여 실행하는 방식
      - `init()`, `start()`, `stop()`, `destroy()`, `paint(Graphics g)`
- 애플릿 응용프로그램 작성에 필요한 클래스
  - ▣ `Applet`, `JApplet` 클래스

# 애플릿이 실행되는 과정

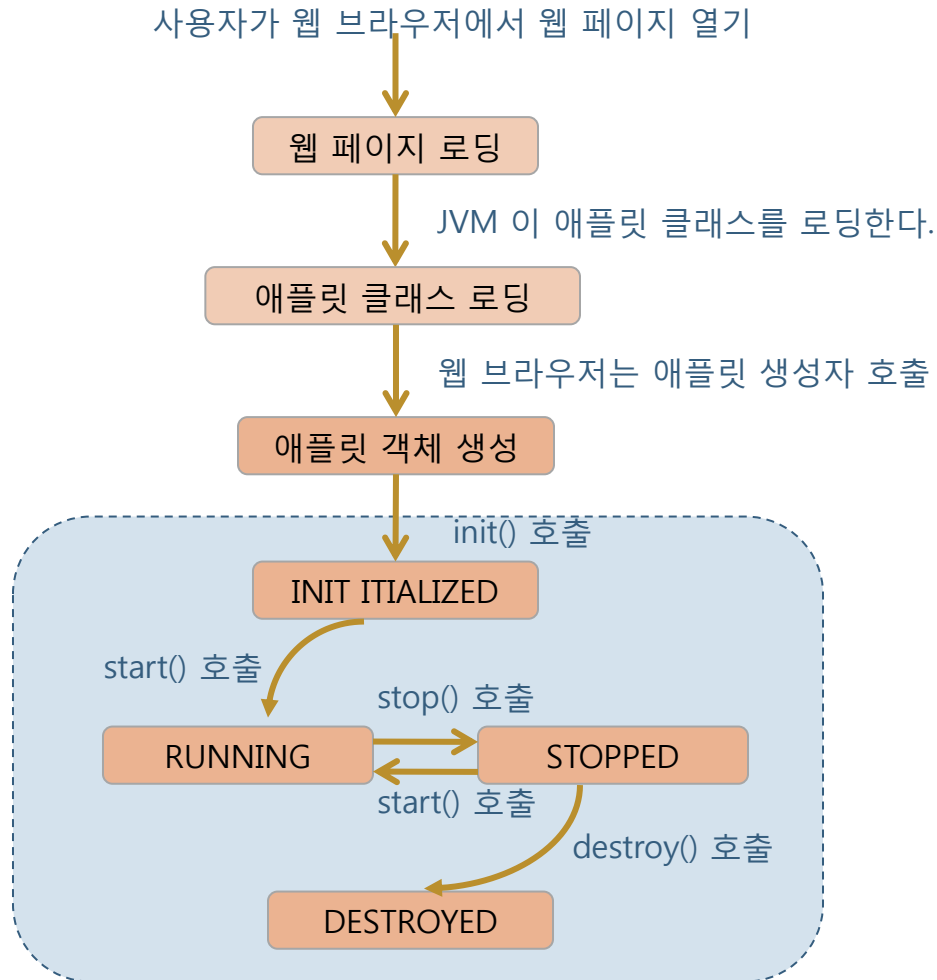
3



- 이미 개발자가 applet.html 파일을 작성하였음
- MyApplet.java 프로그램을 개발하여 MyApplet.class 파일로 컴파일하였음.

# 웹 브라우저에 의한 애플릿의 실행 과정

4



# Applet 클래스와 JApplet 클래스

5

## □ Applet 클래스

- ▣ java.applet.Applet
- ▣ AWT로 작성하는 애플릿 응용프로그램의 최상위 컨테이너
- ▣ 애플릿 작성
  - Applet을 상속받은 클래스 작성
  - Applet에 속한 메소드를 필요에 따라 오버라이딩하여 구현

## □ JApplet 클래스

- ▣ javax.swing.JApplet
- ▣ 스윙으로 작성하는 애플릿 응용프로그램의 최상위 컨테이너
- ▣ 애플릿 작성
  - JApplet을 상속받은 클래스 작성
  - Applet에 속한 메소드를 오버라이딩하여 구현
  - JApplet의 contentPane에 스윙 컴포넌트 부착

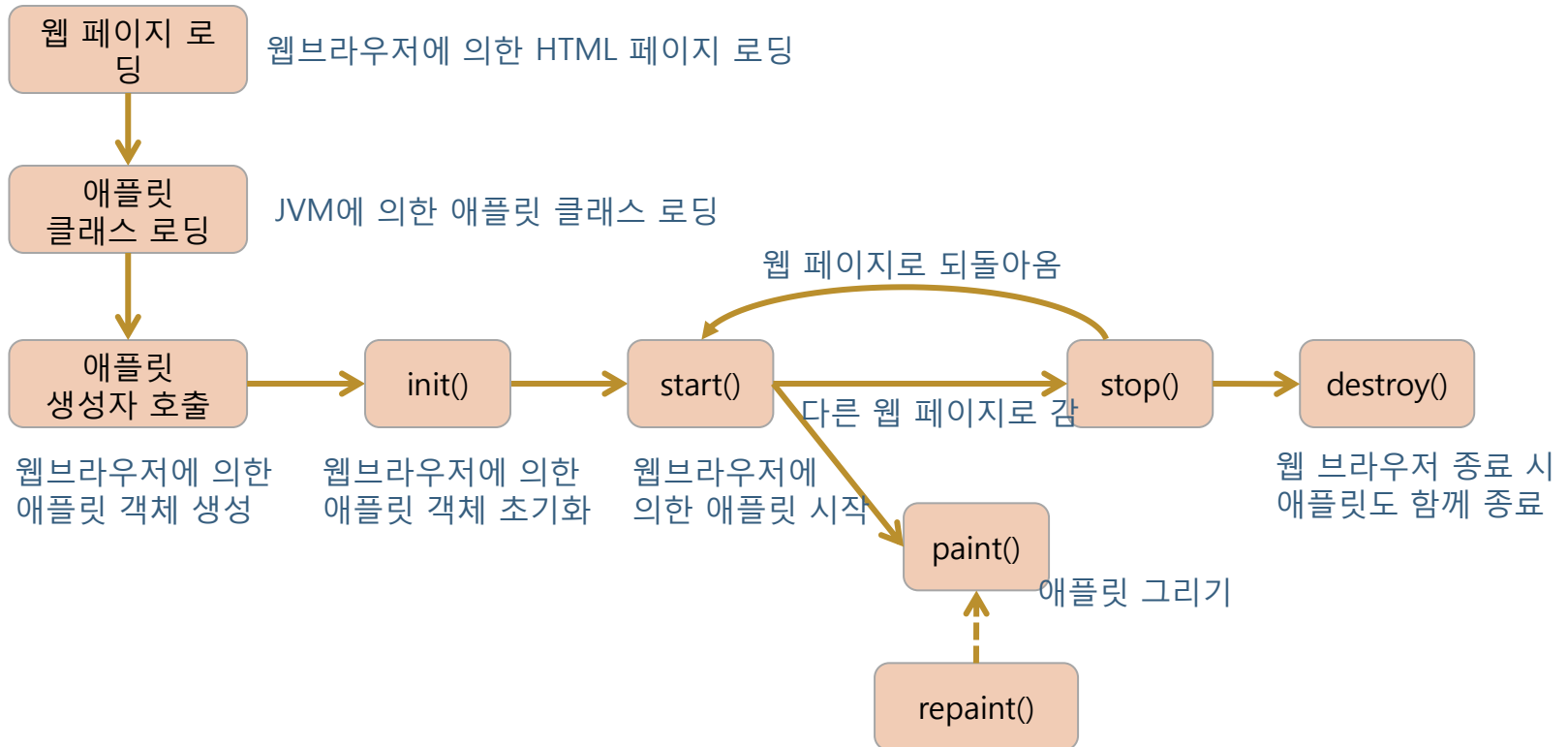
# Applet 클래스로 AWT 애플릿 구현

6

```
public class MyApplet extends Applet {  
    public MyApplet() { ... } // 애플릿을 포함하는 웹 페이지가 로딩된 후 호출되는 생성자  
  
    public void init() { ... } // 애플릿이 처음 로드될 때 호출되는 메소드  
  
    public void start() { ... } // 애플릿을 포함하는 웹 페이지를 방문할 때마다 호출되는 메소드  
  
    public void stop() { ... } // 애플릿을 포함하는 웹 페이지가 비활성화될 때 호출되는 메소드  
  
    public void destroy() { ... } // 웹브라우저가 종료될 때 호출되는 메소드  
  
    public void paint(Graphics g) { ... } // 애플릿을 그리는 메소드  
}
```

# 애플릿의 생명 주기

7



# JApplet 클래스로 스윙 애플릿 구현

8

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

public class MyJApplet extends JApplet {
    public MyJApplet() { ... } // 생성자

    public void init() { ... } // 애플릿이 처음 로드될 때 호출

    public void start() { ... } // 애플릿을 포함하는 웹 페이지 방문 때마다 호출

    public void stop() { ... } // 애플릿을 포함하는 웹 페이지가 비활성화될 때 호출

    public void destroy() { ... } // 웹 브라우저가 종료될 때 호출

    // 스윙 애플릿의 경우 JApplet의 콘텐츠팬의 paintComponent()를 오버라이딩하여
    // 페인팅을 하는 것이 쉽다.
    public void paint(Graphics g) { ... } // paint()는 오버라이딩하지 않는 것이 좋다.
}
```



# HTML 파일과 <applet> 태그

9

```
<applet  
  code=애플릿클래스파일이름  
  width=애플릿이 출력되는 윈도우의 폭  
  height=애플릿이 출력되는 윈도우의 높이  
  [codebase=애플릿의 URL]  
  [alt = 대체 문자열]>  
</applet>
```

```
<applet  
  code="MyApplet.class"  
  width="300"  
  height="300"  
  codebase="appletclasses/"  
  alt ="애플릿을 로딩하지 못하였습니다">  
</applet>
```

## □ applet 태그의 속성

### □ code

- 애플릿 클래스 파일 이름

### □ width, height

- 애플릿이 실행될 윈도우의 폭과 높이
- 픽셀 단위

### □ codebase

- 애플릿은 HTML 파일과 같은 경로에 있지 않는 경우 클래스 파일의 디렉터리
- 애플릿이 HTML과 다른 서버에 있는 경우 애플릿의 URL

### □ alt

- code에 지정된 애플릿 클래스 파일을 로드하지 못한 경우 대신 출력된 문자열

### □ 기타 속성 있음

# 애플릿 만들기

10

## 1. 애플릿 개발

- ▣ AWT로 만드는 애플릿 : Applet 상속
- ▣ 스윙으로 만드는 애플릿 : JApplet 상속

## 2. html 파일 작성

## 3. 애플릿 테스트

- ▣ appletviewer를 이용한 테스트
  - 이클립스에서 실행
  - 콘솔창에서 직접 appletviewer.exe를 직접 실행
- ▣ 웹 브라우저를 이용한 테스트
  - html 문서를 로딩하도록 웹 브라우저 직접 실행
    - 웹 브라우저에 의해 *html* 문서에 내장된 애플릿이 로딩되어 실행됨

# Applet로 만드는 애플릿

11

## MyApplet.html

```
<html>
  <head>
    <title> 애플릿 테스트입니다.</title>
  </head>
  <body>
    <applet code="MyAppletEx.class"
            width="300"
            height="300">
    </applet>
  </body>
</html>
```

애플릿을 내장하는 HTML 파일

## MyAppletEx.java

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

public class MyAppletEx extends Applet {
    String text=null;
    int x;
    int y;
    int fontSize;

    public void init() {
        text = "Hello. It's Applet";
        x = 30;
        y = 30;
        fontSize = 20;
    }

    public void start() {}
    public void stop() {}
    public void destroy() {}

    public void paint(Graphics g) {
        g.setColor(Color.YELLOW);
        g.fillRect(0,0, getWidth(), getHeight());
        g.setColor(Color.RED);
        g.setFont(new Font("Arial", Font.ITALIC, fontSize));
        g.drawString(text, x, y);
    }
}
```

# 애플릿 실행 - appletviewer를 이용한 테스트

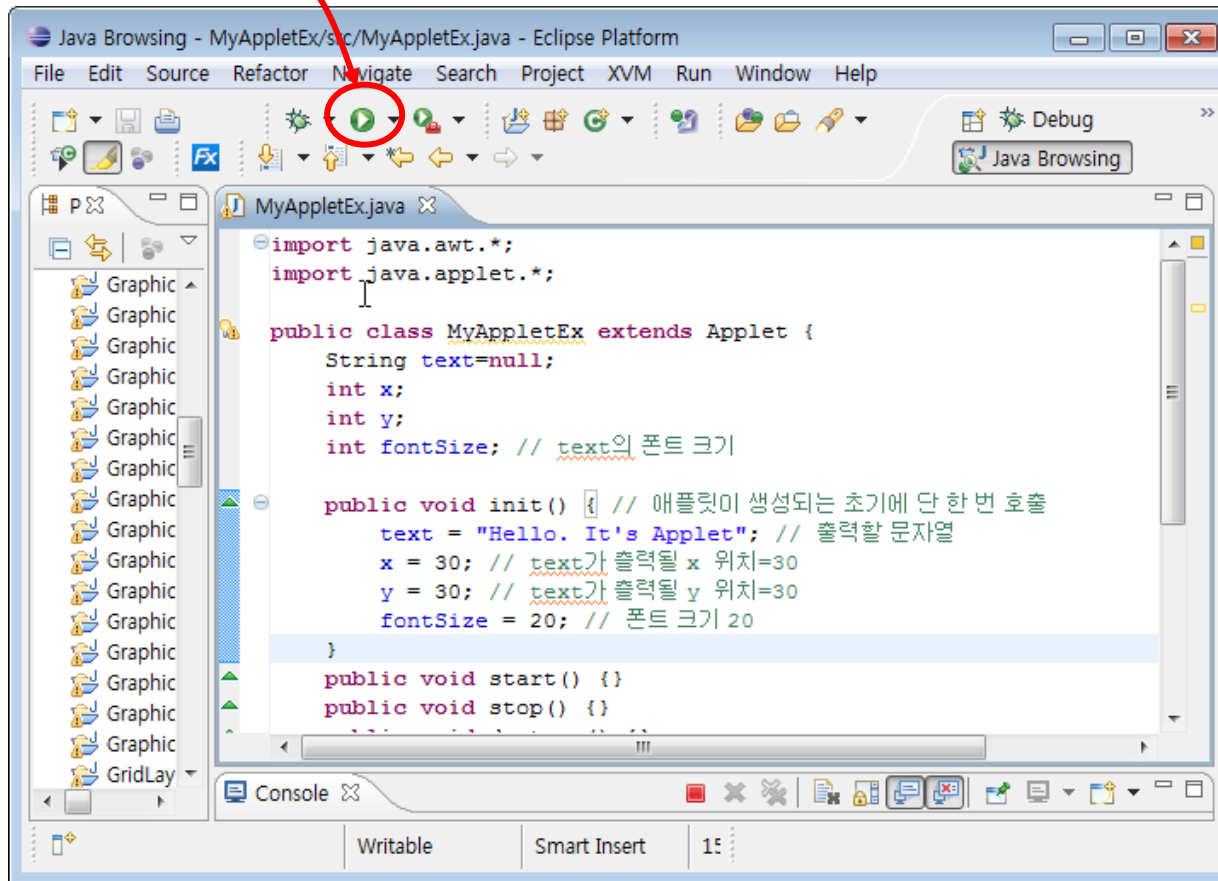
12

- 애플릿 뷰어(appletviewer.exe)
  - ▣ 오라클에서 작성되어 JDK에 배포되는 유틸리티
    - JDK가 설치된 디렉터리 밑의 bin 디렉터리에 있음
  - ▣ 명령창에서 실행 시킬 수 있으나 불편하므로 이클립스에서 바로 실행 가능
    - 자바 프로그램 실행 버튼 클릭

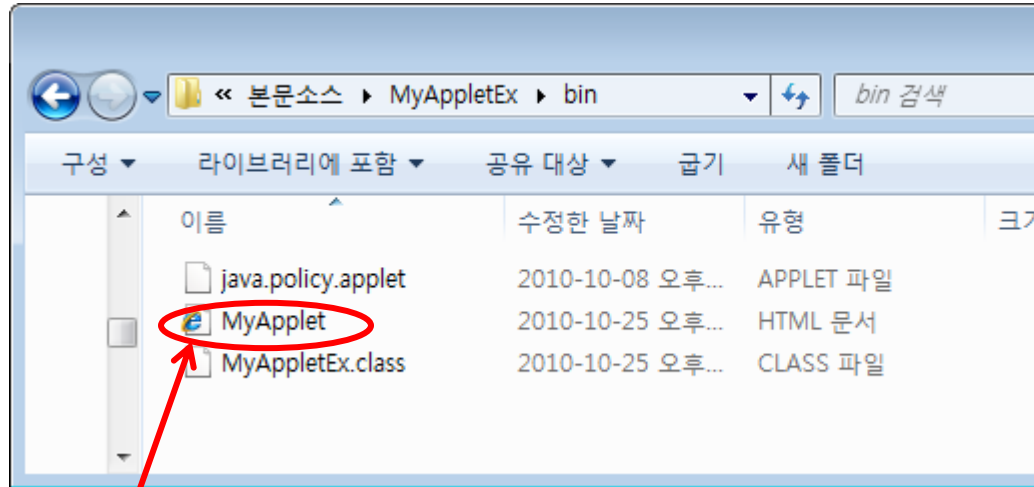
# 이클립스에서 애플릿뷰어로 애플릿 실행

13

애플릿 실행



# 웹 브라우저를 이용한 애플릿 실행

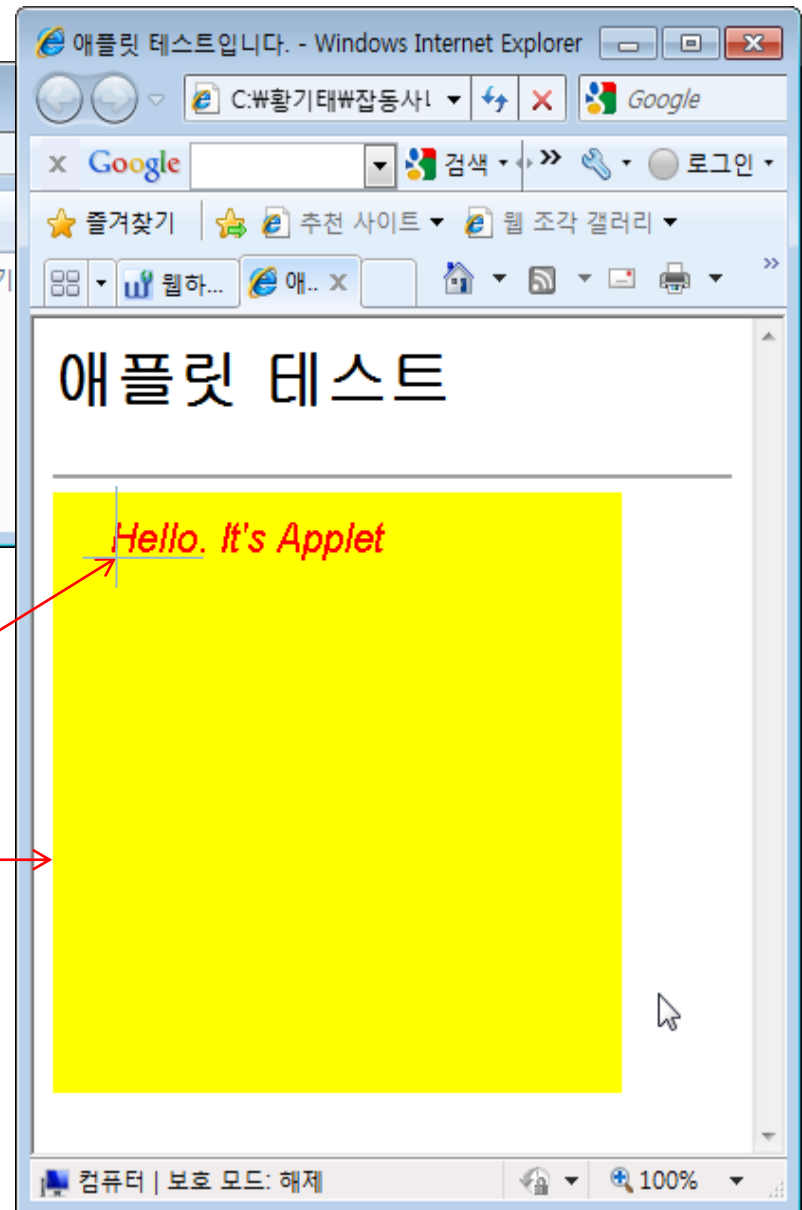


더블클릭하여 웹 브라우저 실행

- MyApplet.html 을 MyAppletEx.class 와 같은 디렉토리에 작성

(30,30)

노란 배경색 영역이  
애플릿이 실행되는  
공간.  
300x300 크기



# 스윙으로 만드는 애플릿

MyJAppletEx.java

MyJApplet.html

```
<html>
  <head>
    <title> 애플릿 테스트입니다.</title>
  </head>
  <body>
    <h1>애플릿 테스트</h1>
    <hr noshade>
    <applet code="MyJAppletEx.class"
            width="300"
            height="300">
    </applet>
  </body>
</html>
```

애플릿을 내장하는 HTML 파일

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

```
public class MyJAppletEx extends JApplet {
```

```
    String text=null;
    int x;
    int y;
    int fontSize;
```

```
    public void init() {
```

```
        text = "Hello. It's Applet";
        x = 30;
        y = 30;
        fontSize = 20;
```

```
        setContentPane(new MyPanel());
```

```
    }
```

```
    public void start() {}
```

```
    public void stop() {}
```

```
    public void destroy() {}
```

```
class MyPanel extends JPanel {
```

```
    public void paintComponent(Graphics g) {
```

```
        super.paintComponent(g);
```

```
        g.setColor(Color.YELLOW);
```

```
        g.fillRect(0,0, getWidth(), getHeight());
```

```
        g.setColor(Color.RED);
```

```
        g.setFont(new Font("Arial", Font.ITALIC, fontSize));
```

```
        g.drawString(text, x, y);
```

```
    }
```

```
}
```

```
}
```

# 애플릿에 파라미터 전달하기

16

## □ 애플릿이 사용자로부터 값을 입력 받기 위한 방법

### ▣ <param> 태그

- <applet> 태그의 내부 태그, name 속성과 value 속성
  - 자바 애플릿 응용프로그램에서 이 *name*으로 지정한 이름으로 *value*의 값 액세스

```
<applet  
  code=애플릿클래스파일이름.  
  width=애플릿이 출력되는 화면 상의 윈도우의 크기.  
  height=애플릿이 출력되는 화면 상의 윈도우의 크기.  
  [codebase=애플릿의 URL]  
  [alt = 대체 문자열] >  
  <param name="파라미터이름1" value="파라미터 값1">  
  <param name="파라미터이름2" value="파라미터 값2">  
  .....  
</applet>
```

## □ 애플릿 응용프로그램에서 파라미터 읽기

### ▣ String Applet.getParameter(String name)

- <param> 태그의 name에 해당하는 value 문자열 리턴



# 애플릿에서 <param> 태그의 파라미터 받기

17

HTML파일

```
<applet code=MyJAppletParamEx.class width=300 height=300>  
  <param name="text" value="Let's study Applet!!!">  
  <param name="xpos" value="10">  
  <param name="ypos" value="100">  
  <param name="fontsize" value="30">  
</applet>
```

애플릿

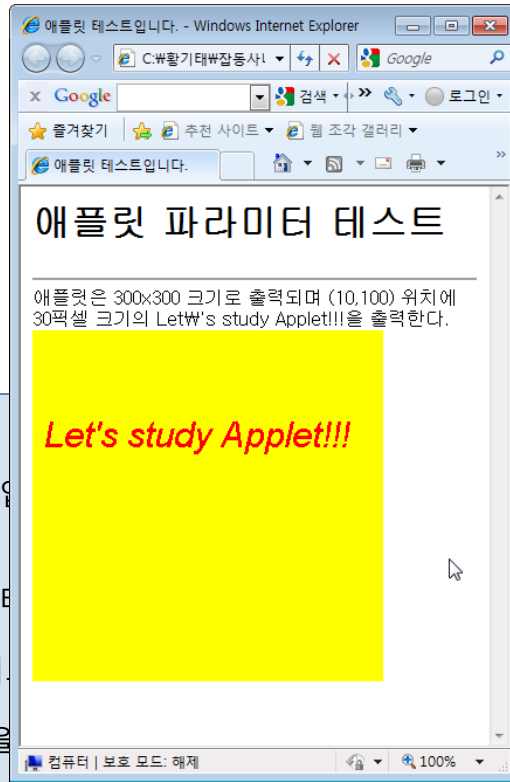
```
public void init() {  
    String text = getParameter("text");  
    int x = Integer.parseInt(getParameter("xpos"););  
    int y = Integer.parseInt(getParameter("ypos"););  
    int fontSize = Integer.parseInt(getParameter("fontsize"););  
}
```

getParameter("text")의 리턴 값은  
"Let's study Applet!!!"

```
text      = "Let's study Applet!!!"  
x         = 10  
y         = 100  
fontSize  = 30
```

init() 메소드의 실행 결과

# <param> 태그로부터 파라미터를 읽는 애플릿 예제



```
<html>
<head>
  <title> 애플릿 테스트</title>
</head>
<body>
  <h1>애플릿 파라미터 테스트</h1>
  <hr noshade>
  애플릿은 300x300 크기
  의
  LetW's study Applet!!!을
  <br>
  <applet code=MyJAppletParamEx.class width=300 height=300>
    <param name="text" value="Let's study Applet!!!">
    <param name="xpos" value="10">
    <param name="ypos" value="100">
    <param name="fontsize" value="30">
  </applet>
</body>
</html>
```

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

```
public class MyJAppletParamEx extends JApplet {
  String text=null;
  int x=0;
  int y=0;
  int fontSize=10;
  public void init() {
    text = getParameter("text");
    try {
      x = Integer.parseInt(getParameter("xpos"));
      y = Integer.parseInt(getParameter("ypos"));
      fontSize =
        Integer.parseInt(getParameter("fontsize"));
    } catch (NumberFormatException e) {}

    setContentPane(new MyPanel());
  }

  class MyPanel extends JPanel {
    public void paintComponent(Graphics g) {
      super.paintComponent(g);
      if(text == null) return;
      g.setColor(Color.YELLOW);
      g.fillRect(0,0, getWidth(), getHeight());
      g.setColor(Color.RED);
      g.setFont(new Font("Arial", Font.ITALIC,
        fontSize));
      g.drawString(text, x, y);
    }
  }
}
```

\* appletviewer로 실행할 수 없음

# 애플릿의 보안에 따른 제약

19

## □ 애플릿의 보안 문제

### ▣ 애플릿은 서버에 있는 코드가 클라이언트 상에서 실행

- 서버 코드가 안전함을 확신하기 어려움
- 애플릿의 클라이언트 컴퓨터에 대한 접근 제약을 둠
  - 예) 애플릿은 클라이언트 컴퓨터의 하드를 접근할 수 없도록 함

### ▣ trusted applet

- 클라이언트 컴퓨터의 파일 시스템에 대한 접근이 허용된 애플릿
- Java2부터 새로운 보안 프로토콜 도입됨
- <http://www.developer.com/java/ent/article.php/3303561>

## □ 애플릿의 제약 사항

### ▣ 클라이언트 파일시스템 접근 불허

- 애플릿은 클라이언트 컴퓨터의 파일을 읽고 쓸 수 없다.

### ▣ 클라이언트 컴퓨터의 타 프로그램 실행 불허

- 애플릿은 클라이언트 컴퓨터 상에 설치된 프로그램을 실행할 수 없다.
- 클라이언트 컴퓨터 상의 정보를 빼오거나 클라이언트 컴퓨터를 망가뜨리거나 파일을 삭제하거나 할 수 있기 때문

### ▣ 네트워크 접속 불허

- 애플릿은 클라이언트 컴퓨터에서 다른 컴퓨터로 네트워크 접속할 수 없다.
- 애플릿이 다운로드되었던 서버하고만 유일하게 통신 가능

# 애플릿에서 오디오 다루기

20

- 오디오 포맷
  - ▣ Wav, AIFF, MIDI, AU, RMF
- 오디오 클립
  - ▣ 재생 가능한 오디오 정보를 담은 객체
- 오디오 재생을 위한 과정
  - ▣ 오디오 클립 객체 생성
    - Java 2 이전 - 애플릿에서만 오디오 재생 가능
    - Java 2 이후 - 데스크톱 응용프로그램에서도 재생 가능

```
Class classObject = this.getClass();  
URL url = classObject.getResource("song.au"); // class 디렉토리에 있는 song.au 의 URL  
AudioClip audioClip = Applet.newAudioClip(url); // 오디오 클립 생성
```

# 오디오 재생

21

## □ AudioClip의 메소드

### ▣ void play()

- 오디오 클립의 연주를 시작한다. 항상 처음부터 시작한다.

### ▣ void stop()

- 오디오 연주를 중단한다.

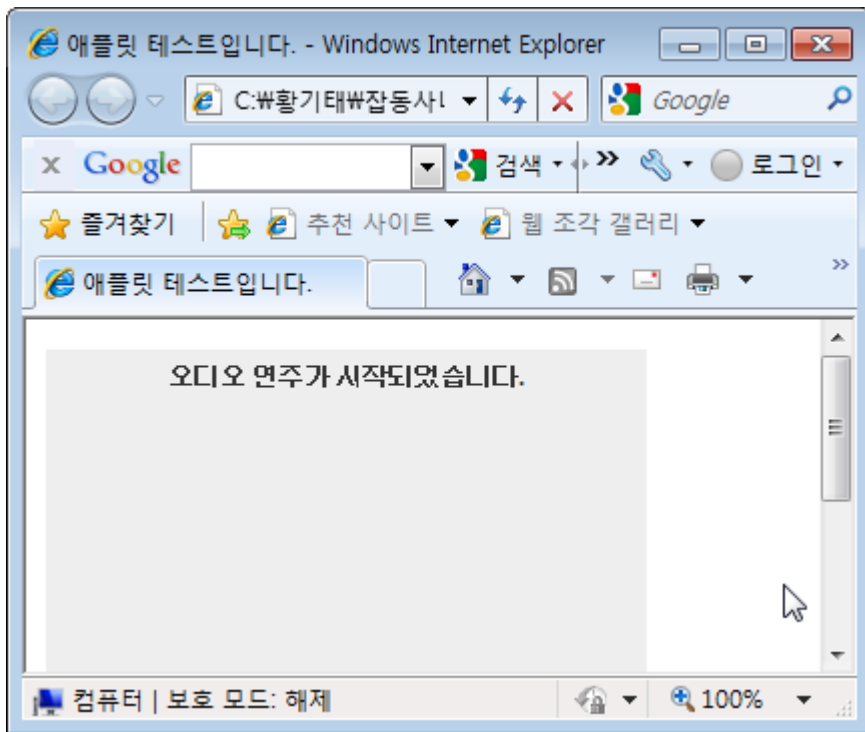
### ▣ void loop()

- 오디오 클립을 반복적으로 연주한다.

# 예제 15-1 : 애플릿에서 오디오 연주하기

AudioJAppletEx.html

```
<html>
<head>
<title> 애플릿 테스트입니다.</title>
</head>
<body>
<applet code=AudioJAppletEx.class width=300 height=300>
</applet>
</body>
</html>
```



AudioJAppletEx.java

```
import java.awt.*;
import javax.swing.*;
import java.applet.*;
import java.net.URL;

public class AudioJAppletEx extends JApplet {
    AudioClip clip=null;
    public void init() {
        setContentPane(new MyPanel());
        URL audioURL = getClass().getResource("ToYou.mid");
        clip = Applet.newAudioClip(audioURL);
    }
    public void start() {
        if(clip != null) {
            clip.play();
            ((MyPanel)getContentPane()).setText("오디오 연주가 시작되었습니다.");
        }
    }
    public void stop() {
        if(clip != null) clip.stop();
    }

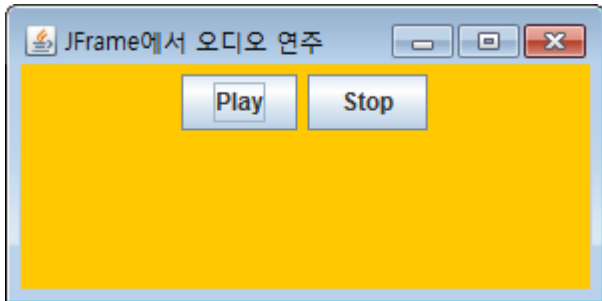
    class MyPanel extends JPanel {
        JLabel label = new JLabel();
        MyPanel() {
            add(label);
        }
        void setText(String text) {
            label.setText(text);
        }
    }
}
```

\*ToYou.mid 파일을  
class 파일이 있는 디  
렉터리에 삽입

# 예제 15-2 : 오디오 재생/중지 가능한 데스크톱 응용프로그램 작성

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.applet.*;
import java.net.URL;

public class AudioFrameEx extends JFrame {
    AudioFrameEx() {
        setTitle("JFrame에서 오디오 연주");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setContentPane(new MyPanel());
        setSize(300, 150);
        setVisible(true);
    }
}
```



Play 버튼을 누르면 ToYou.mid 연주  
Stop 버튼을 누르면 연주 중단

```
class MyPanel extends JPanel {
    AudioClip clip = null;
    JButton btn[] = new JButton [2];

    MyPanel() {
        setBackground(Color.ORANGE);
        setLayout(new FlowLayout());
        MyActionListener listener = new MyActionListener();
        btn[0] = new JButton("Play");
        btn[1] = new JButton("Stop");
        for(int i=0; i<btn.length; i++) {
            add(btn[i]);
            btn[i].addActionListener(listener);
        }

        URL audioURL = getClass().getResource("ToYou.mid");
        clip = Applet.newAudioClip(audioURL);
    }

    class MyActionListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            if(e.getActionCommand().equals("Play"))
                clip.play();
            else
                clip.stop();
        }
    }
}

public static void main(String[] args) {
    new AudioFrameEx();
}
```