{ JAVA }

선린인터넷
웹 운 영 과

# 그래픽 사용자 인터페이스 : GUI

AWT Swing

# Graphical User Interface 맛보기

AWT Swing

## 미리 알아두기 1

### ❖ GUI 객체 타입

- AWT : Abstract Windows Toolkit – 자바 초기 버전
- 스윙(Swing) : 스윙에 속하는 클래스는 클래스 이름 앞에 대문자 **J** 가 붙여짐
  - **컴포지트(Composite) 디자인 패턴**을 사용해서 설계되었음

### ❖ GUI 패키지

- java.awt : GUI 컴포넌트의 부모클래스, Color나 Point 등 **유틸리티** 클래스
- java.awt.event : **이벤트 처리** 클래스 또는 인터페이스
- javax.swing : GUI **컨테이너** 또는 **컴퍼넌트** 클래스

## 미리 알아두기 2

### ❖ Container 컨테이너 클래스

- 컴퍼넌트를 추가, 배치할 수 있는 역할을 하며, 논리적으로 그룹화함
- 컨테이너는 다른 컨테이너나 컴퍼넌트를 포함할 수 있는 계층구조를 가짐
- 예: JFrame JWindow JDialog Japple Jpanel 등

### ❖ Component 컴퍼넌트 클래스

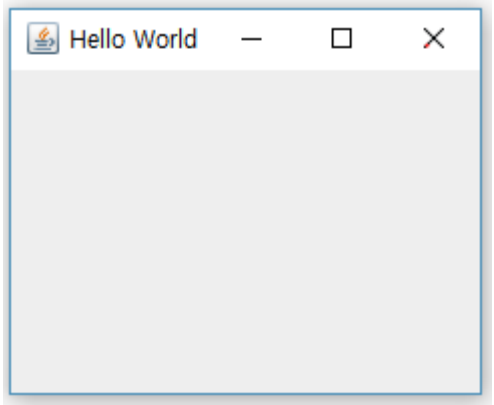
- 컨테이너 위에 붙여서 사용자와 컴퓨터 간의 상호작용하는 그래픽 요소
- 컨테이너에 붙여야만 화면에 보일 수 있음
- 예: JButton JLabel 등

# 첫번째

# 윈도우 창 띄우기

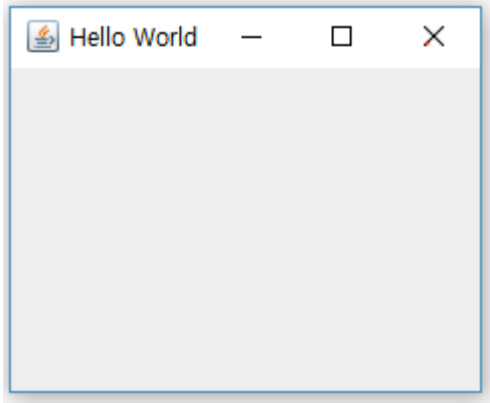
컨테이너 생성

컨테이너 생성
JFrame클래스



## Exam1 첫번째 : 윈도우 창 띄우기

```java
// swing클래스들은 javax.swing패키지에 있다.
import javax.swing.JFrame;

public class Exam1 {
    public static void main(String[] args) {
        // new연산자를 이용해서 JFrame객체를 생성한다.
        // 참조변수 frame이 객체를 참조한다.
        JFrame frame = new JFrame();
        frame.setSize(250, 200); // 프레임 크기를 설정한다.

        // 프레임 오른쪽 상단 X버튼을 누르면, 프로그램을 종료토록 설정한다.
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("Hello World"); // 프레임 타이틀 제목을 설정한다.
        frame.setLocation(200, 200); // 프레임 위치를 설정한다.

        frame.setVisible(true); // 프레임을 화면에 나타나게 한다.
    }
}
```



# 2번째

# 윈도우창에 버튼 추가

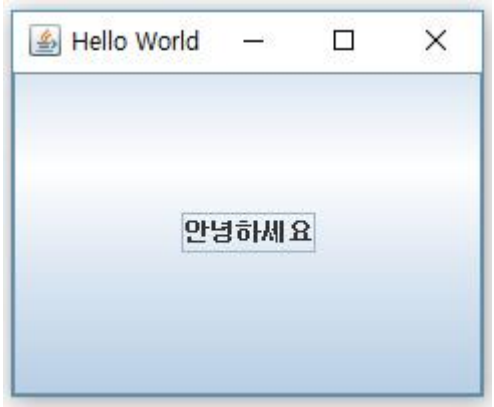
1 컨테이너 생성

2 컴퍼넌트 추가

# GUI 맛보기

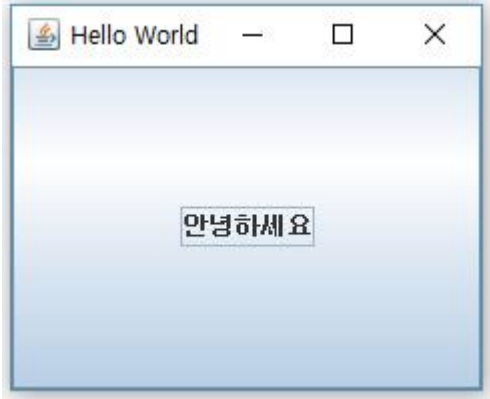
1 컨테이너 생성
JFrame클래스

2 컴퍼넌트 추가
JButton클래스



## Exam1 2번째 : 버튼 1개 붙이기

```java
import javax.swing.JButton;
// swing클래스들은 javax.swing패키지에 있다.
import javax.swing.JFrame;

public class Exam1 {
    public static void main(String[] args) {
        // new연산자를 이용해서 JFrame 컨테이너 객체를 생성한다.
        // 참조변수 frame이 객체를 참조한다.
        JFrame frame = new JFrame();
        frame.setSize(250, 200); // 프레임 크기를 설정한다.

        // 프레임 오른쪽 상단 X버튼을 누르면, 프로그램을 종료토록 설정한다.
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("Hello World"); // 프레임 타이틀 제목을 설정한다.
        frame.setLocation(200, 200); // 프레임 위치를 설정한다.

        // Button 컴포넌트 객체를 생성한다.
        JButton button = new JButton("안녕하세요");
        frame.add(button); // frame 컨테이너에 button객체를 배치한다.

        frame.setVisible(true); // 프레임을 화면에 나타나게 한다.
    }
}
```



# 3번째

**ActionListener** 인터페이스 활용

# 이벤트 처리를 해보자1

## 버튼을 누르면, 콘솔창에 출력하기

이벤트 소스 (컴포넌트 객체) → 이벤트 객체가 발생됨 → 이벤트 리스너 객체 (메소드)에 전달됨

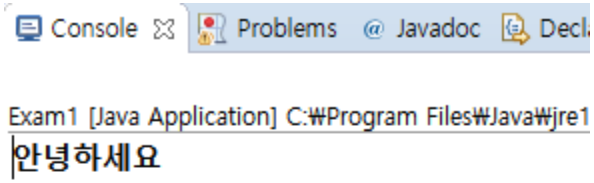
1 컨테이너 생성

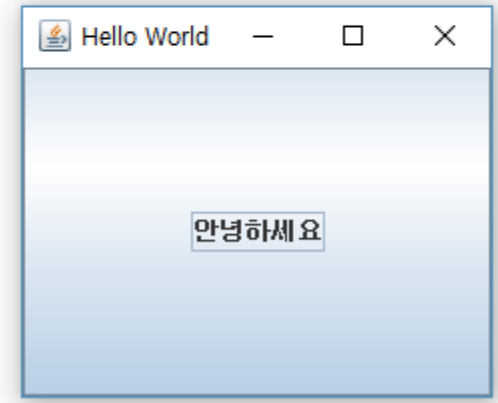
2 컴퍼넌트 추가

3 이벤트 리스너
클래스를 작성

4 컴퍼넌트에
**리스너 객체**를
등록

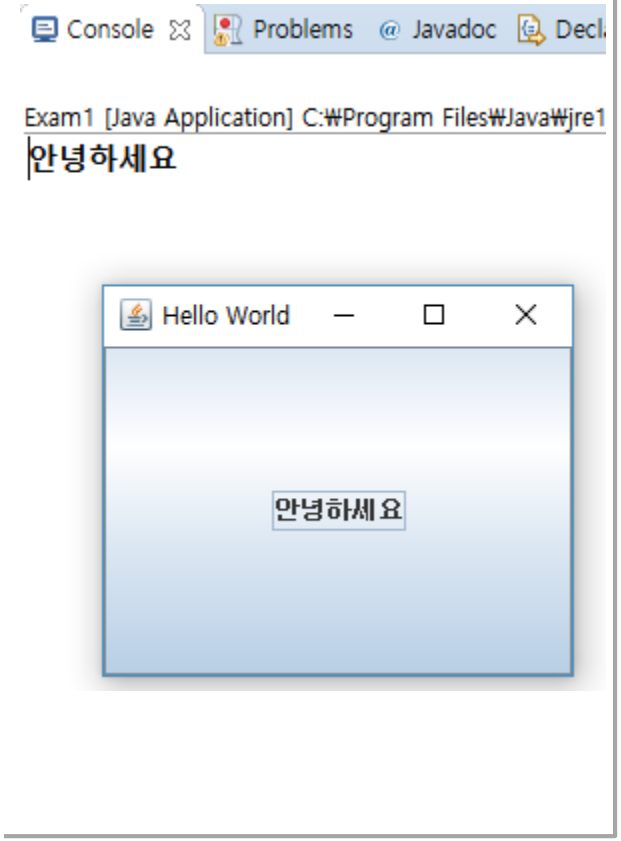
getSource( ) 및
getText( ) 메소드



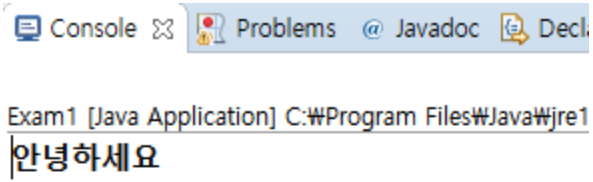
## Exam1 3번째 : 버튼을 누르면, 콘솔창에 버튼의 텍스트 내용을 출력하기

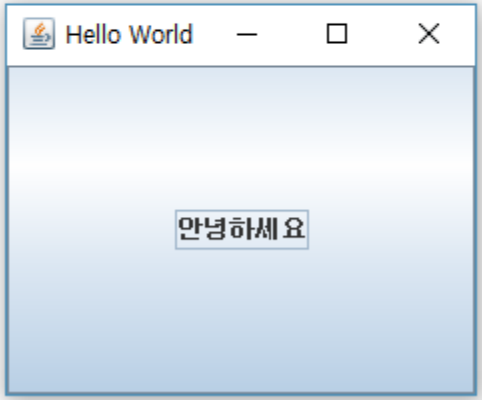
```
1  // swing클래스들은 javax.swing패키지에 있다.
2  import javax.swing.*;
3  public class Exam1 {
4      public static void main(String[] args) {
5          // new연산자를 이용해서 JFrame 컨테이너 객체를 생성한다.
6          // 참조변수 frame이 객체를 참조한다.
7          JFrame frame = new JFrame();
8          frame.setSize(250, 200); // 프레임 크기를 설정한다.
9
10         // 프레임 오른쪽 상단 X버튼을 누르면, 프로그램을 종료토록 설정한다.
11         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
12         frame.setTitle("Hello World"); // 프레임 타이틀 제목을 설정한다.
13         frame.setLocation(200, 200); // 프레임 위치를 설정한다.
14
15         // Button 컴포넌트 객체를 생성한다.
16         JButton button = new JButton("안녕하세요");
17         // 이벤트 리스터 클래스를 작성해 객체를 생성해서, 컴포넌트 객체에 등록한다.
18         // 즉, 이벤트 리스너 객체를 생성하고, 버튼에 이벤트 리스너 객체를 등록한다.
19         button.addActionListener(new MyListener());
20         frame.add(button); // frame 컨테이너에 button객체를 배치한다.
21         frame.setVisible(true); // 프레임을 화면에 나타나게 한다.
22     }
23 }
```



## Exam1 3번째 : 버튼을 누르면, 콘솔창에 버튼의 텍스트 내용을 출력하기

```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;

public class MyListener implements ActionListener {
    // 이벤트 객체는 발생된 이벤트에 대한 모든 정보를 리스너에 전달한다.
    // 이벤트 객체는 getSource()메소드를 가지는데,
    // 이벤트를 발생한 이벤트 소스 객체를 Object타입으로 반환한다.
    // 따라서 이 객체를 필요한 타입으로 형변환해서 사용하면 된다.
    @Override
    public void actionPerformed(ActionEvent e) {
        // getSource()메소드를 통해 이벤트 소스 객체를 얻는다.
        JButton button = (JButton) e.getSource();

        // getText() : button객체의 텍스트 정보를 얻어온다.
        System.out.println(button.getText());
    }
}
```



ActionListener 인터페이스 활용

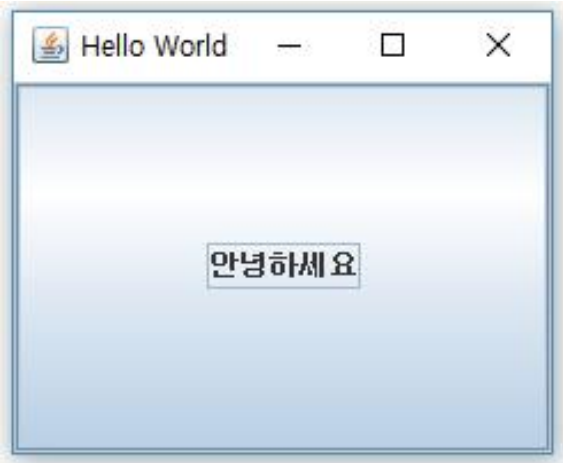
# 4번째

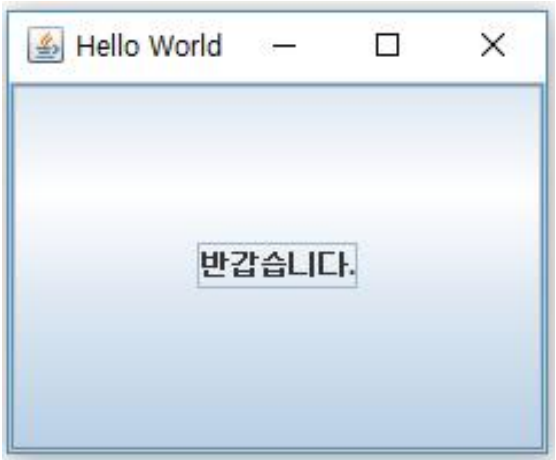
# 이벤트 처리를 해보자2

안녕하세요 ⟷ 반갑습니다

힌트: setText( ) 메소드를 이용해보자

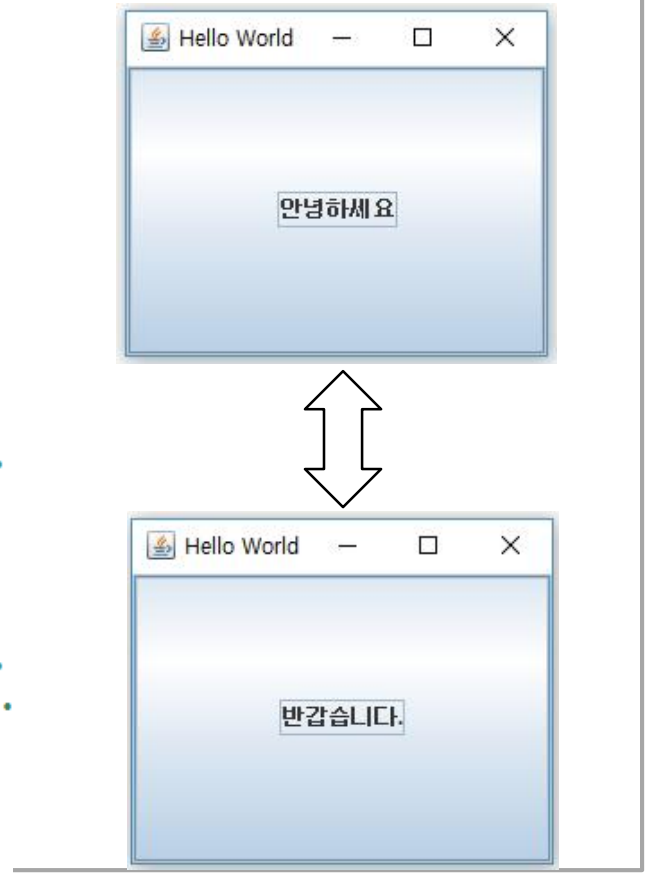
# 안녕하세요 ⟷ 반갑습니다



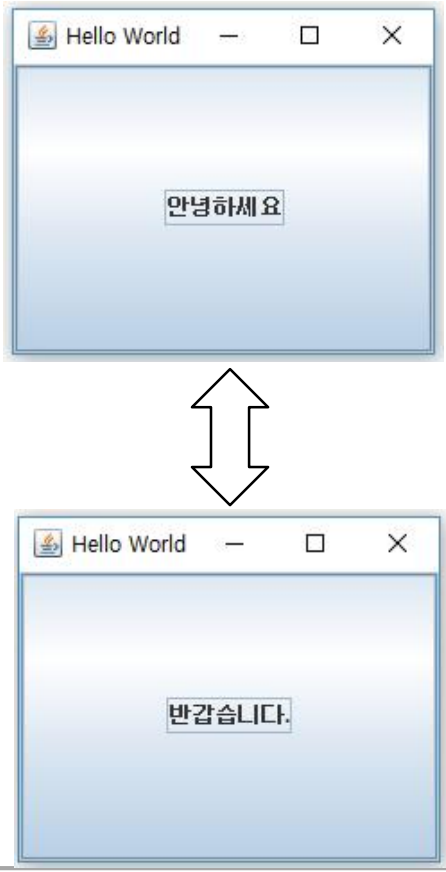
## Exam1 4번째 : 버튼을 누르면, 텍스트를 바뀌도록 해보자 (이벤트 처리)

```java
// swing클래스들은 javax.swing패키지에 있다.
import javax.swing.*;
public class Exam1 {
    public static void main(String[] args) {
        // new연산자를 이용해서 JFrame 컨테이너 객체를 생성한다.
        // 참조변수 frame이 객체를 참조한다.
        JFrame frame = new JFrame();
        frame.setSize(250, 200); // 프레임 크기를 설정한다.

        // 프레임 오른쪽 상단 X버튼을 누르면, 프로그램을 종료토록 설정한다.
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("Hello World"); // 프레임 타이틀 제목을 설정한다.
        frame.setLocation(200, 200); // 프레임 위치를 설정한다.

        // Button 컴포넌트 객체를 생성한다.
        JButton button = new JButton("안녕하세요");
        // 이벤트 리스터 클래스를 작성해 객체를 생성해서, 컴포넌트 객체에 등록한다.
        // 즉, 이벤트 리스너 객체를 생성하고, 버튼에 이벤트 리스너 객체를 등록한다.
        button.addActionListener(new MyListener());
        frame.add(button); // frame 컨테이너에 button객체를 배치한다.
        frame.setVisible(true); // 프레임을 화면에 나타나게 한다.
    }
}
```



## Exam1 4번째 : 버튼을 누르면, 텍스트를 바뀌도록 해보자 (이벤트 처리)

```java
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
public class MyListener implements ActionListener {
    // 이벤트 객체는 발생된 이벤트에 대한 모든 정보를 리스너에 전달한다.
    // 이벤트 객체는 getSource()메소드를 가지는데,
    // 이벤트를 발생한 이벤트 소스 객체를 Object타입으로 반환한다.
    // 따라서 이 객체를 필요한 타입으로 형변환해서 사용하면 된다.
    @Override
    public void actionPerformed(ActionEvent e) {
        // getSource()메소드를 통해 이벤트 소스 객체를 얻는다.
        JButton button = (JButton) e.getSource();

        // 누를 때마다 "안녕하세요" <--> "반갑습니다"로 변경한다.
        if(button.getText().equals("안녕하세요"))
            button.setText("반갑습니다.");
        else
            button.setText("안녕하세요");
    }
}
```



# 5번째

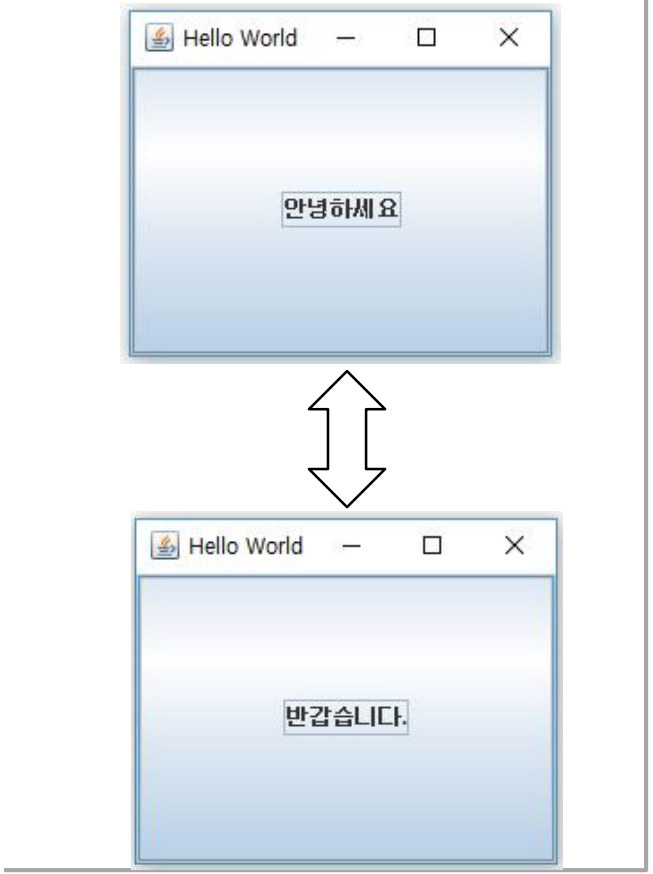
**MouseListener** 인터페이스 활용
(실행 내용은 같게 합시다.)

# 이벤트 처리를 해보자3

**안녕하세요 ⟷ 반갑습니다**

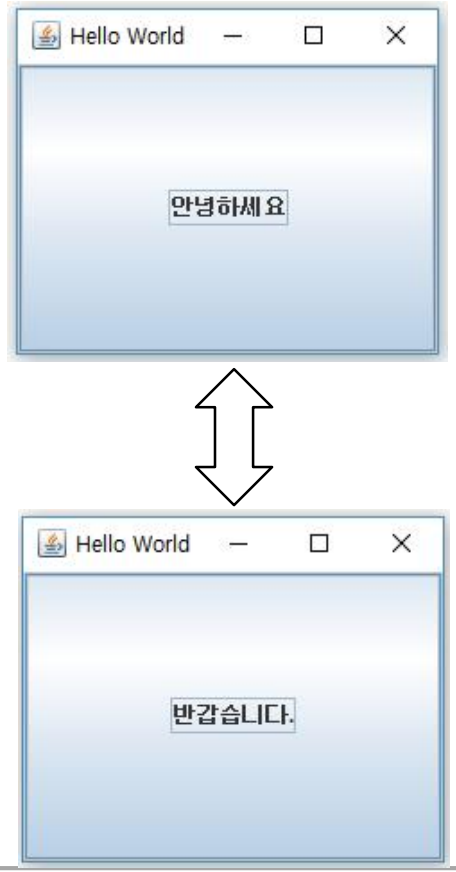
## Exam1 5번째 : 버튼을 누르면, 텍스트를 바뀌도록 해보자 (이벤트 처리)

```java
// swing클래스들은 javax.swing패키지에 있다.
import javax.swing.*;
public class Exam1 {
    public static void main(String[] args) {
        // new연산자를 이용해서 JFrame 컨테이너 객체를 생성한다.
        // 참조변수 frame이 객체를 참조한다.
        JFrame frame = new JFrame();
        frame.setSize(250, 200); // 프레임 크기를 설정한다.

        // 프레임 오른쪽 상단 X버튼을 누르면, 프로그램을 종료토록 설정한다.
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setTitle("Hello World"); // 프레임 타이틀 제목을 설정한다.
        frame.setLocation(200, 200); // 프레임 위치를 설정한다.

        // Button 컴포넌트 객체를 생성한다.
        JButton button = new JButton("안녕하세요");
        // 이벤트 리스터 클래스를 작성해 객체를 생성해서, 컴포넌트 객체에 등록한다.
        // 즉, 이벤트 리스너 객체를 생성하고, 버튼에 이벤트 리스너 객체를 등록한다.
        button.addMouseListener(new MyMouseListener());//리스너 교체
        frame.add(button); // frame 컨테이너에 button객체를 배치한다.
        frame.setVisible(true); // 프레임을 화면에 나타나게 한다.
    }
}
```



## Exam1 5번째 : 버튼을 누르면, 텍스트를 바꿔도록 해보자 (이벤트 처리)

```
1  import java.awt.event.MouseEvent;
2  import java.awt.event.MouseListener;
3  import javax.swing.JButton;
4  public class MyMouseListener implements MouseListener {
5      public void mouseClicked(MouseEvent e) {
6          JButton button = (JButton) e.getSource();
7          if(button.getText().equals("안녕하세요"))
8              button.setText("반갑습니다.");
9          else
10             button.setText("안녕하세요");
11     }
12     public void mouseEntered(MouseEvent e) { }
13     public void mouseExited(MouseEvent e) { }
14     public void mousePressed(MouseEvent e) { }
15     public void mouseReleased(MouseEvent e) { }
16 }
```



# 다음 시간은...

GUI 계속 …

화 이 팅 !