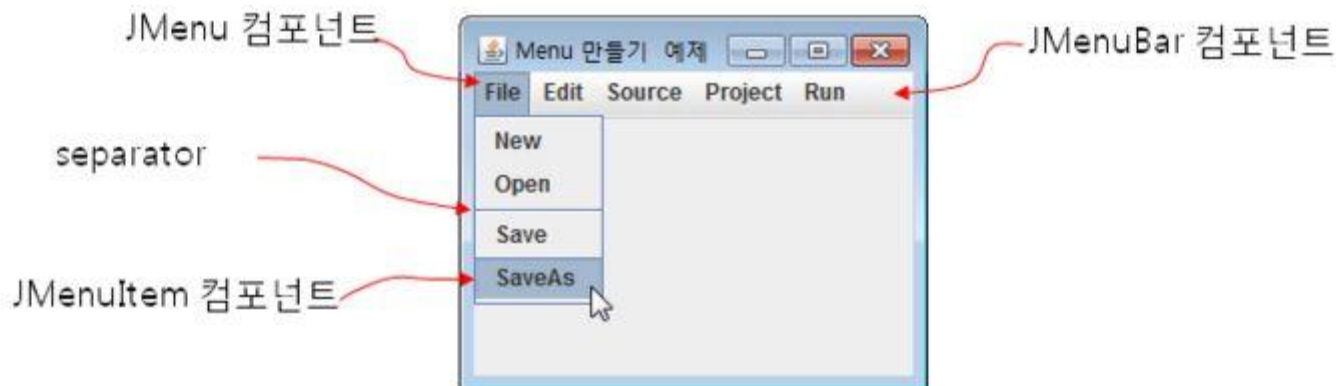


# 메뉴 만들기

2

- 메뉴 만들기에 필요한 스윙 컴포넌트
  - JMenuBar
    - 메뉴바의 기능을 하는 컴포넌트
  - JMenu
    - 파일, 편집 등 하나의 메뉴 기능을 하는 컴포넌트
  - JMenuItem
    - 파일 메뉴 내에 저장 등의 세부 기능을 하는 컴포넌트



# 메뉴 만드는 과정

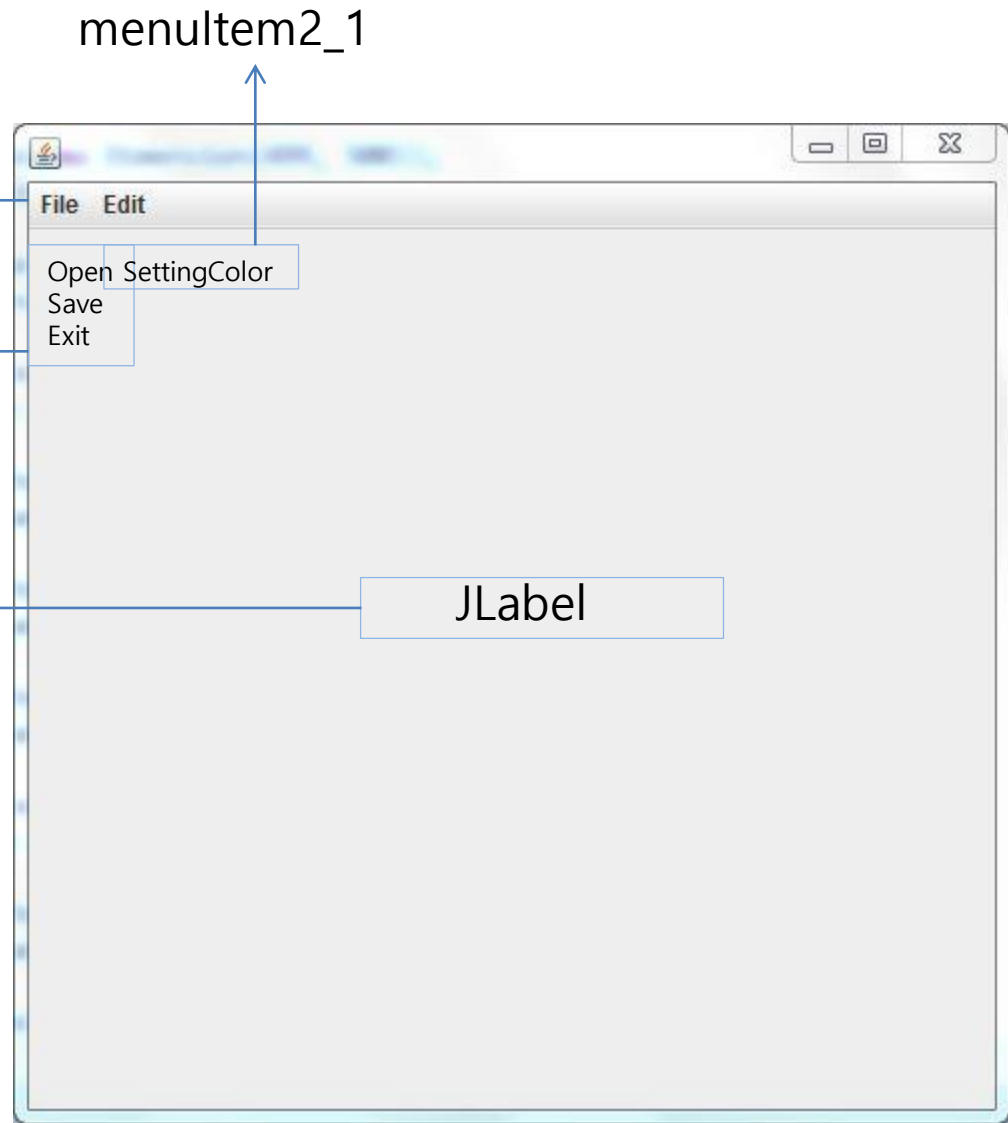
3



파워 빌더를 이용하여  
다음과 같은 메뉴바를  
만들어 보자

menuBar  
(menu1, menu2)  
( menuItem1\_1  
menuItem1\_2  
menuItem1\_3 )

imageLabel



# 파일 다이얼로그

24

## □ JFileChooser

- 파일 시스템의 탐색기와 같은 기능을 하는 다이얼로그
- 사용자에게 파일이나 디렉터리를 쉽게 선택하도록 하는 기능
- 다이얼로그 종류
  - 파일 열기 다이얼로그(File Open Dialog)
  - 파일 저장 다이얼로그(File Save Dialog)

## □ 주의 사항

- 파일 다이얼로그에서 사용자가 파일을 선택하는 행위란?
  - 사용자가 선택한 파일 이름을 얻기 위함
  - 파일을 읽거나 쓰는 행위가 아님
- 다이얼로그 닫기란?
  - 다이얼로그가 화면에서 보이지 않게 됨.
  - JFileChooser 객체가 없어지거나 사용할 수 없게 것이 아님
    - `showOpenDialog()` 메소드를 호출하면 다시 화면에 출력하고 재사용 가능
  - 다이얼로그가 닫힌 후 JFileChooser 객체로부터 사용자가 입력한 정보를 알아낼 수 있음

# JFileChooser 파일 다이얼로그, 코드 샘플

## 파일 열기 다이얼로그 생성 과정

1. JFileChooser 객체 생성

```
JFileChooser chooser = new JFileChooser();
```

2. 파일 필터  
객체 생성

```
FileNameExtensionFilter filter = new FileNameExtensionFilter(  
    "JPG & GIF",  
    "jpg", "gif");
```

3. JFileChooser에  
파일 필터 설정

```
chooser.setFileFilter(filter);
```

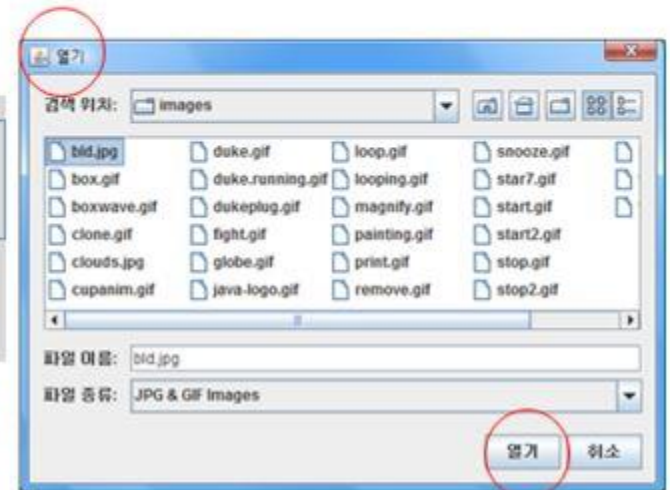
생략가능

4. 열기 다이얼로그 출력

```
intret = chooser.showOpenDialog(null);
```

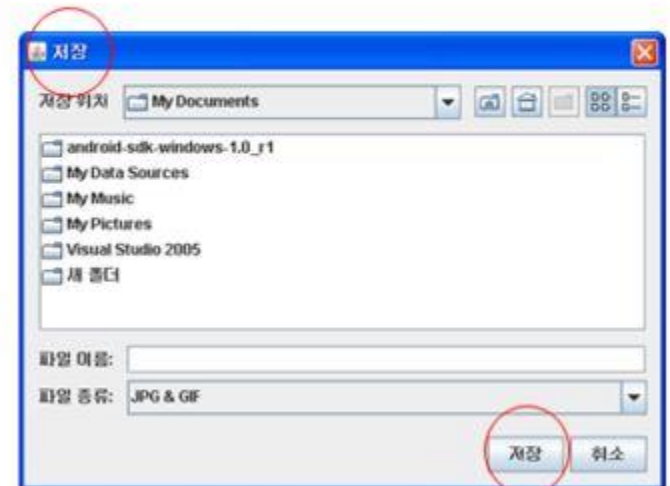
5. 사용자가 선택한  
파일 이름 알아내기

```
if(ret == JFileChooser.APPROVE_OPTION) {  
    String pathName = chooser.getSelectedFile().getPath();  
    String fileName = chooser.getSelectedFile().getName();  
}
```



## 파일 저장 다이얼로그 생성 과정

```
intret = chooser.showSaveDialog(null);
```



```
public class MenuEx extends JFrame {
```

```
    JLabel imageLabel = new JLabel();
```

```
    public MenuEx() {  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        JMenuBar menuBar = new JMenuBar();  
        setJMenuBar(menuBar);
```

```
        JMenu menu1 = new JMenu("File");  
        menuBar.add(menu1);
```

```
        JMenuItem menuItem1 = new JMenuItem("open");  
        menuItem1.addActionListener(new OpenListener());  
        menu1.add(menuItem1);
```

```
        JMenuItem menuItem2 = new JMenuItem("save");  
        menu1.add(menuItem2);
```

```
        JMenuItem menuItem3 = new JMenuItem("exit");  
        menu1.add(menuItem3);
```

```
        JMenu menu2 = new JMenu("Edit");  
        menuBar.add(menu2);
```

```
        JMenuItem menuItemSettingColor = new JMenuItem("setting Color");  
        menu2.add(menuItemSettingColor);
```

```
        getContentPane().add(imageLabel, BorderLayout.CENTER);
```

```
        setSize(new Dimension(499, 500));  
        setVisible(true);  
    }
```

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

```
class OpenListener implements ActionListener {
```

```
    JFileChooser chooser = new JFileChooser();
```

```
    @Override
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        FileNameExtensionFilter filter =
```

```
            new FileNameExtensionFilter("JPG & Gif Images","jpg","gif");
```

```
        chooser.setFileFilter(filter);
```

```
        // 파일 다이얼로그 출력
```

```
        int ret = chooser.showOpenDialog(null);
```

```
        // 사용자가 파일을 선택하고 "열기" 버튼을 누른 경우
```

```
        String filePath = chooser.getSelectedFile().getPath(); //파일 경로명을 알아온다.
```

```
        // 파일을 로딩하여 이미지 레이블에 출력한다.
```

```
        imageLabel.setIcon(new ImageIcon(filePath));
```

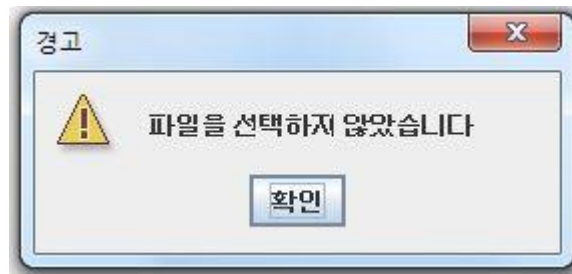
```
        pack(); // 이미지의 크기에 맞추어 프레임의 크기 조절
```

```
    }
```

```
}
```

위의 예에서

파일다이얼로그에서 사용자가 취소 버튼을 눌렀을 경우 콘솔에 에러가 출력된다. 사용자가 취소버튼을 눌렀을 경우 팝업창으로 경고메세지를 띄우자





# 팝업 다이얼로그, JOptionPane

20

## □ JOptionPane

- 간단하고 유용한 팝업 다이얼로그 지원
- static 의 타입의 간단한 메소드 이용. 간단히 다이얼로그 출력 가능

## □ 입력 다이얼로그 - 한 줄을 입력 받는 다이얼로그

- `String JOptionPane.showInputDialog(String msg)`
  - `msg`: 다이얼로그의 객체
  - 리턴값: 사용자가 입력한 문자열

## □ 확인 다이얼로그 - 사용자로부터 Yes/No 응답을 입력 받는 다이얼로그

- `int JOptionPane.showConfirmDialog(Component parentComponent, Object msg, String title, int optionType)`
  - `parentComponent`는 다이얼로그가 출력되는 영역의 범위로서 프레임, null 이면 화면 전체의 중앙에 출력
  - `msg`: 다이얼로그에 출력될 객체
  - `title`: 다이얼로그의 타이틀 문자열
  - `optionType`: 다이얼로그의 종류 지정 - YES\_ON\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTION
  - 리턴값: 사용자가 선택한 옵션 - YES\_OPTION, NO\_OPTION, CANCEL\_OPTION, OK\_OPTION, CLOSED\_OPTION

## □ 메시지 다이얼로그 - 단순 메시지를 출력하는 다이얼로그

- `void JOptionPane.showMessageDialog(Component parentComponent, Object msg, String title, int messageType)`
  - `parentComponent, msg, title`은 `showConfirmDialog()` 메소드와 동일
  - `messageType` - ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, PLAIN\_MESSAGE

```
class OpenListener implements ActionListener {  
    JFileChooser chooser = new JFileChooser();
```

```
@Override
```

```
public void actionPerformed(ActionEvent e) {
```

```
    FileNameExtensionFilter filter =
```

```
        new FileNameExtensionFilter("JPG & Gif Images","jpg","gif");  
    chooser.setFileFilter(filter);
```

```
    int ret = chooser.showOpenDialog(null);
```

```
    // 사용자가 창을 강제로 닫았거나 취소 버튼을 누른 경우
```

```
    if(ret != JFileChooser.APPROVE_OPTION) {
```

```
        JOptionPane.showMessageDialog(null, "파일을 선택하지 않았습니다",  
            "경고", JOptionPane.WARNING_MESSAGE);
```

```
        return;
```

```
    }
```

```
    // 사용자가 파일을 선택하고 "열기" 버튼을 누른 경우
```

```
    String filePath = chooser.getSelectedFile().getPath();
```

```
    imageLabel.setIcon(new ImageIcon(filePath));
```

```
    pack();
```

```
}
```

리스너등록 잊지 마세요!!!

```
menuItem1_1.addActionListener(  
    new OpenListener());
```

```

class SaveListener implements ActionListener {
    JFileChooser chooser = new JFileChooser();

    @Override
    public void actionPerformed(ActionEvent e) {

        FileNameExtensionFilter filter = new
            FileNameExtensionFilter("JPG Images","jpg");
        chooser.setFileFilter(filter);

        int ret = chooser.showSaveDialog(null);

        if(ret == JFileChooser.APPROVE_OPTION) {
            String filePath = chooser.getSelectedFile().getPath();
            File saveFile = new File(filePath);
            //이미지 공간 확보하기
            BufferedImage newImage = new BufferedImage(imageLabel.getIcon().getIconWidth(),
                imageLabel.getIcon().getIconHeight(), BufferedImage.TYPE_INT_RGB);

            Graphics g = newImage.getGraphics();
            imageLabel.getIcon().paintIcon(null, g, 0, 0);//생성된 공간에 그리기(복사)
            g.dispose(); 만들어진 그래픽 삭제
            try {
                ImageIO.write(newImage, "jpg", saveFile);//파일에 저장하기
            }catch (IOException ex) {
            }
        }
    }
}

```

```
class ExitListener implements ActionListener {  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        System.exit(0);  
    }  
}
```

# 컬러 다이얼로그

28

## □ JColorChooser

### ▣ 색상 팔레트를 제공하는 모달 다이얼로그

#### 컬러 다이얼로그 생성 과정

##### 1. JColorChooser 객체 생성

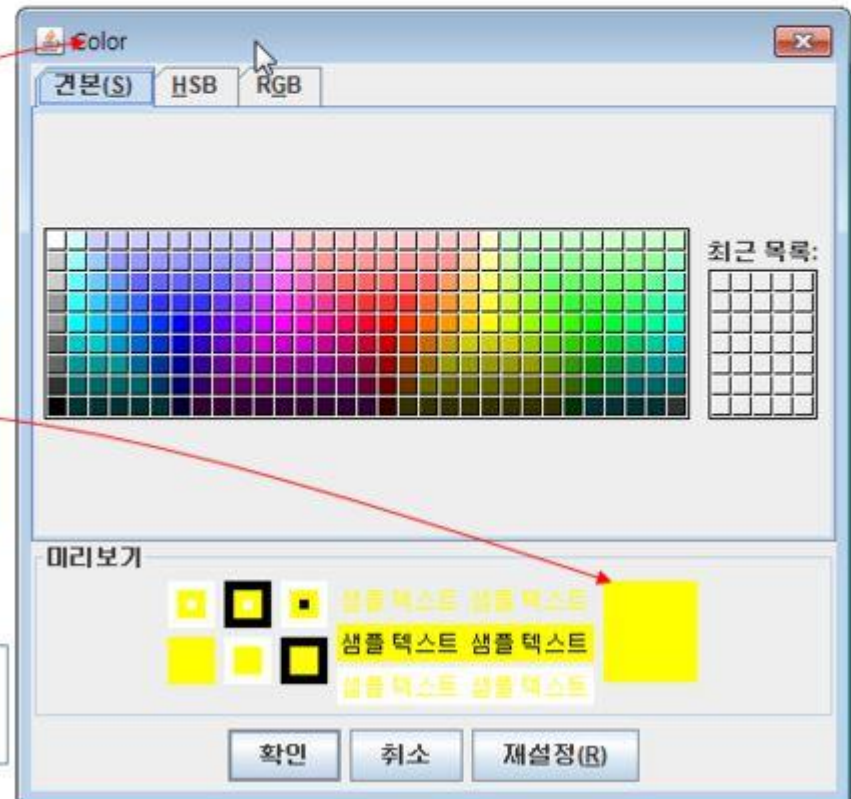
```
JColorChooser chooser = new JColorChooser();
```

##### 2. 컬러 다이얼로그 출력

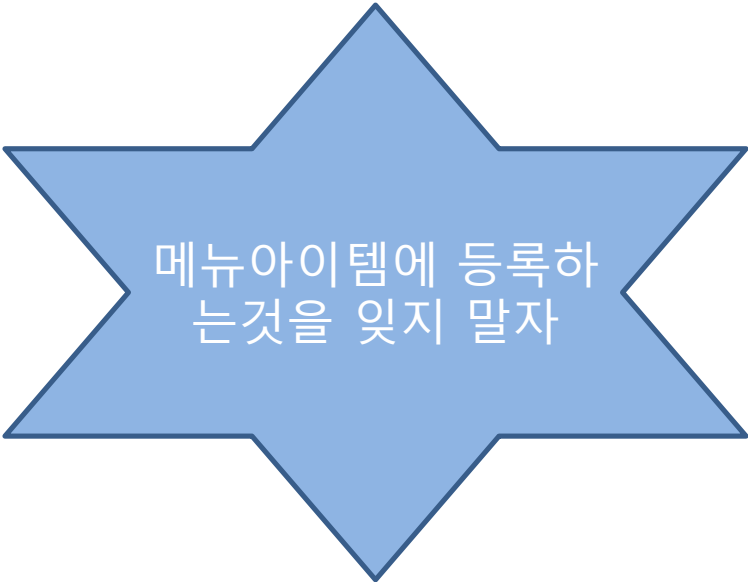
```
Color selectedColor = chooser.showDialog(null,  
    "Color",  
    Color.YELLOW);  
// 사용자가 "확인" 버튼을 누르면 선택한 색을 리턴하지만  
// 사용자가 "취소" 버튼이나 다이얼로그를 강제로 닫을 때  
// null 리턴
```

##### 3. 사용자가 색을 선택하였는지 확인

```
if(selectedColor != null) {  
    // 사용자가 선택한 색 selectedColor 사용  
}
```



```
class ColorListener implements ActionListener {  
    JColorChooser chooser = new JColorChooser();  
  
    @Override  
    public void actionPerformed(ActionEvent e) {  
  
        Color selectedColor =  
            chooser.showDialog(null, "Color", Color.BLUE);  
  
    }  
}
```



메뉴아이템에 등록하  
는것을 잊지 말자