

Cheat Sheet based on:

Swing/GUI Cheat Sheet at Williams College

<http://eventfuljava.cs.williams.edu/s04/handouts/SwingGUICheatSheet.html>

Displaying a Swing component

Construct and initialize the component.

```
button = new JButton ("Bu tto nLa bel ");
```

Add it to the content pane of the window or to a JPanel that is added to the display.

```
getCon ten tPa ne( ).add (button);
```

Import javax.swing. *and sometimes also java.awt.* at the beginning of the class creating the components.

```
import javax.s wing.; import java.awt.
```

Getting events from GUI component

Declare that the class handling the event implements the appropriate listener interface.

```
implements Action Lis tener
```

Define the method that the listener interface requires.

```
public void action Per formed (Actio nEvent event)
```

Add a listener appropriate for the component to the component.

```
button.ad dAc tio nLi stener (this);
```

Import java.awt.event. (*and occasionally javax.swing.event.*) at the beginning of the class that is the listener.

```
import javax.s wing.; import java.awt.
```

Finding out which component sent the event

When the listener method is called, you can find out which component sent the event by calling getSource() on the event:

```
public void action Per formed (Actio nEvent event)
{
    Object theButton = event.getSource();
    if (theButton == framed Cir cle Button) {
        // Create a framed circle
    }
}
```

If a method returns a String, remember to compare the result using the equals method, not ==:

```
aMenu.g et Sel ect edI tem ().e quals ("A value");
```

Containers

JPanel constr-uctor: `new JPanel ()`

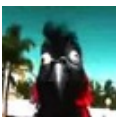
Define the type of layout: `void setLayout (Layou tMa nager lm)`

Add an object to a container: (FlowLayout or GridLayout) `void add (Component c)`

Add an object to a container: (BorderLayout) `void add (Component c, int position)`

Both JPanel and the object obtained by sending getContentPane() to a WindowController object are containers (and have type Container). These methods are available for all containers.

For BorderLayouts, position may be either BorderLayout.NORTH, BorderLayout.SOUTH, BorderLayout.EAST, BorderLayout.WEST, or BorderLayout.CENTER.



By NeonKnightOA

Published 11th November, 2015.

Last updated 12th May, 2016.

Page 1 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

Layout Managers

BorderLayout constructor: `new BorderLayout ()`

FlowLayout constructor: `new FlowLayout ()`

GridLayout constructor: `new GridLayout (int rows, int cols, int colSpa cing, int rowSpa cing)`

BorderLayout is the default layout for WindowController, whereas FlowLayout is default for JPanel.

GUI Components - General

The following methods can be applied to any Component:

`void setFont (Font f)`
`void setFor eground (Color c)`
`void setBac kground (Color c)`

To construct a font use:

`new Font (String name, int style, int size)`

Style can be one of the following:

`Font.BOLD`
`Font.ITALIC`
`Font.PLAIN`
`Font.BOLD+Font.ITALIC`

GUI Components - JButton

Constr-uctor: `new JButton (String s)`

General `String getText ()`
Methods: `void setText (String s)`

Listener `Action Lis tener`
Interface:

Adding the listener: `void addAct ion Lis tener (Actio nLi stene r al)`

Listening Method: `void action Per formed (Actio nEvent e)`

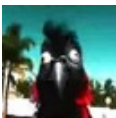
GUI Components - JComboBox

Constr-uctor and Initializ-ation: `new JComboBox ()`
`void addItem (Object item)`

General `Object getSel ect edItem ()`
Methods: `String text=(String)menu.getSelectedItem();`
`int getSel ect edIndex ()`

Listener `ItemLi stener`
Interface: `Action Lis tener`

Adding the listener: `void addIte mLi stener (ItemL istener il)`
`void addAct ion Lis tener (Actio nLi stener al)`



By NeonKnightOA

Published 11th November, 2015.
 Last updated 12th May, 2016.
 Page 2 of 3.

Sponsored by **Readable.com**
 Measure your website readability!
<https://readable.com>

GUI Components - JComboBox (cont)

Listening Method: `void itemState Changed (ItemEvent e)`
`void actionPerformed (ActionEvent e)`

About methods:

`getSelectedItem ( )` returns the selected item
`(String) menu.getSelectedItem ( )`; is a typecast which treats the above returned value as a String
`int getSelectedItemIndex ( )` returns the index of the selected item.

About the listeners:

This component can hear the user making a menu selection depending on the chosen interface. Be consistent in your choice of listener interface, adding method, and listening method.

GUI Components - JLabel

Constructors: `new JLabel (String s)`
`new JLabel (String s, int align)`

General Methods: `void setText (String s)`
`String getText ( )`

Listener Interface: No listeners available.

align can be either `JLabel.RIGHT`, `JLabel.LEFT` or `JLabel.CENTER`.

GUI Components - JSlider

Constructor: `new JSlider (int orientation, int minimum, int maximum, int initialValue)`

General Methods: `void setValue (int newVal)`
`int getValue ( )`

Listener Interface: `ChangeListener`

Adding the Listener: `addChangeListener (ChangeListener al)`

Listening Method: `void stateChanged (ChangeEvent e)`

orientation can be either `JSlider.HORIZONTAL` or `JSlider.VERTICAL`.

GUI Components - JTextField

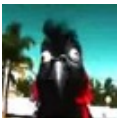
Constructors: `new JTextField (String s)`

General Methods: `void setText (String s)`
`String getText ( )`

Listener Interface: `ActionListener`

Adding the Listener: `addActionListener (ActionListener al)`

Listening Method: `void actionPerformed (ActionEvent e)`



By NeonKnightOA

Published 11th November, 2015.

Last updated 12th May, 2016.

Page 3 of 3.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>