

# JComboBox

JComboBox es una parte del paquete Java Swing. JComboBox hereda la clase JComponent. JComboBox muestra un menú emergente que muestra una lista y el usuario puede seleccionar una opción de esa lista especificada. JComboBox puede ser editable o leído, solo dependiendo de la elección del programador.

**Los constructores de JComboBox son:**

1. **JComboBox()** : crea un nuevo JComboBox vacío.
2. **JComboBox(ComboBoxModel M)**: crea un nuevo JComboBox con elementos de ComboBoxModel especificado
3. **JComboBox(E [ ] i)** : crea un nuevo JComboBox con elementos de la matriz especificada.
4. **JComboBox(Vector items)** : crea un nuevo JComboBox con elementos del vector especificado

**Los métodos más utilizados son:**

1. **addItem(E item)** : agrega el elemento a JComboBox
2. **addItemListener( ItemListener l)** : agrega un ItemListener a JComboBox
3. **getItemAt(int i)**: devuelve el elemento en el índice i
4. **getItemCount()**: devuelve el número de elementos de la lista
5. **getSelectedItem()** : devuelve el elemento seleccionado
6. **removeItemAt(int i)** : quita el elemento en el índice i
7. **setEditable(boolean b)** : el booleano **b** determina si el cuadro combinado es editable o no. Si se pasa true, el cuadro combinado es editable o viceversa.
8. **setSelectedIndex(int i)**: selecciona el elemento de JComboBox en el índice i.
9. **showPopup()** :hace que el cuadro combinado muestre su ventana emergente.
10. **setUI(ComboBoxUI ui)**: establece el objeto L&F que representa este componente.
11. **setSelectedItem(Object a)**: establece el elemento seleccionado en el área de visualización del cuadro combinado en el objeto del argumento.
12. **setSelectedIndex(int a)**: selecciona el elemento en index anIndex.
13. **setPopupVisible(boolean v)**: establece la visibilidad de la ventana emergente.
14. **setModel(ComboBoxModel a)**: establece el modelo de datos que JComboBox utiliza para obtener la lista de elementos.
15. **setMaximumRowCount(int count)**: establece el número máximo de filas que muestra JComboBox.
16. **setEnabled(boolean b)**: habilita el cuadro combinado para que se puedan seleccionar elementos.
17. **removeItem(Object anObject)**: elimina un elemento de la lista de elementos.
18. **removeAllItems()**: elimina todos los elementos de la lista de elementos.
19. **removeActionListener(ActionListener l)**: quita un ActionListener.
20. **isPopupVisible()** : determina la visibilidad de la ventana emergente.
21. **addPopupMenuListener(PopupMenuListener l)**: agrega un agente de escucha PopupMenu que escuchará los mensajes de notificación de la parte emergente del cuadro combinado.

22. **getActionCommand()** : devuelve el comando action que se incluye en el evento enviado a los agentes de escucha de acciones.
23. **getEditor()**: devuelve el editor utilizado para pintar y editar el elemento seleccionado en el campo JComboBox.
24. **getItemCount()** : devuelve el número de elementos de la lista.
25. **getItemListeners()**: devuelve una matriz de todos los ItemListeners agregados a este JComboBox con **addItemListener()**.
26. **createDefaultKeySelectionManager()** : devuelve una instancia del administrador de selección de claves predeterminado.
27. **fireItemStateChanged(ItemEvent e)**: notifica a todos los agentes de escucha que han registrado interés en la notificación de este tipo de evento.
28. **firePopupMenuCanceled()**: notifica a **PopupMenuListeners** que la parte emergente del cuadro combinado se ha cancelado.
29. **firePopupMenuWillBecomeInvisible()** : notifica a **PopupMenuListeners** que la parte emergente del cuadro combinado se ha vuelto invisible.
30. **firePopupMenuWillBecomeVisible()** : notifica a **PopupMenuListeners** que la parte emergente del cuadro combinado será visible.
31. **setEditor(ComboBoxEditor a)**: establece el editor utilizado para pintar y editar el elemento seleccionado en el campo JComboBox.
32. **setActionCommand(String a)**: establece el comando action que debe incluirse en el evento enviado a **actionListeners**.
33. **getUI()**: devuelve el objeto de aspecto que representa este componente.
34.  **paramString()** : devuelve una representación de cadena de este JComboBox.
35. **getUIClassID()**: devuelve el nombre de la clase Look and feel que representa este componente.
36. **getAccessibleContext()** : obtiene el **AccessibleContext** asociado a este JComboBox

## Los siguientes programas ilustrarán el uso de JComboBox

### 1. Programa para crear un JComboBox simple y agregarle elementos.

```
// Java Program to create a simple JComboBox
// and add elements to it
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
class solve extends JFrame implements ItemListener {

    // frame
    static JFrame f;

    // label
    static JLabel l, l1;

    // combobox
    static JComboBox c1;

    // main class
```

```
public static void main(String[] args)
{
    // create a new frame
    f = new JFrame("frame");

    // create a object
    solve s = new solve();

    // set layout of frame
    f.setLayout(new FlowLayout());

    // array of string containing cities
    String s1[] = { "Jalpaiguri", "Mumbai", "Noida", "Kolkata", "New Delhi" };

    // create checkbox
    c1 = new JComboBox(s1);

    // add ItemListener
    c1.addItemListener(s);

    // create labels
    l = new JLabel("select your city ");
    l1 = new JLabel("Jalpaiguri selected");

    // set color of text
    l.setForeground(Color.red);
    l1.setForeground(Color.blue);

    // create a new panel
    JPanel p = new JPanel();

    p.add(l);

    // add combobox to panel
    p.add(c1);

    p.add(l1);

    // add panel to frame
    f.add(p);

    // set the size of frame
    f.setSize(400, 300);

    f.show();
}

public void itemStateChanged(ItemEvent e)
```

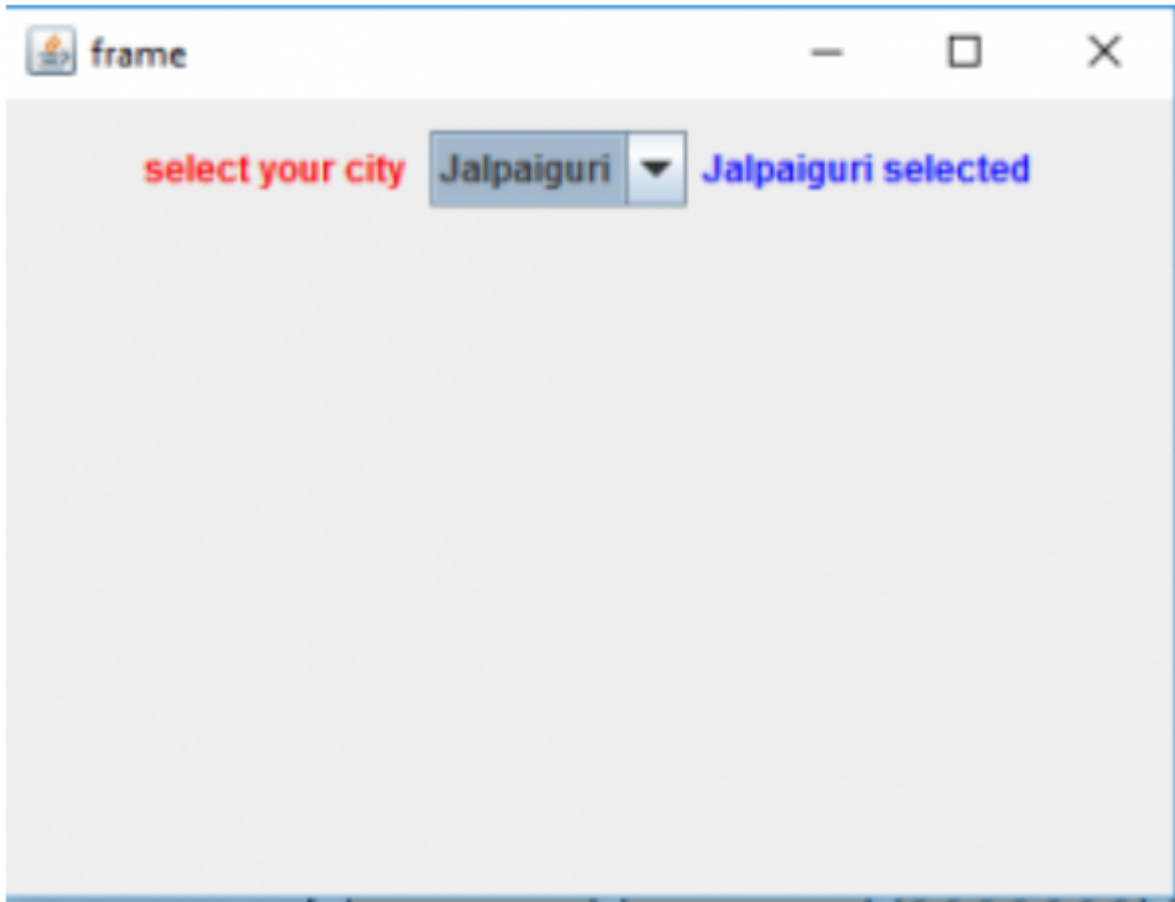
```

{
    // if the state combobox is changed
    if (e.getSource() == c1) {

        l1.setText(c1.getSelectedItem() + " selected");
    }
}
}

```

**Salidad:**



## 2. Programa para crear dos casillas de verificación, una editable y otra de solo lectura

```

// Java Program to create two checkbox
// one editable and other read only
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
class solve extends JFrame implements ItemListener {

    // frame
    static JFrame f;

```

```
// label
static JLabel l, l1, l3, l4;

// combobox
static JComboBox c1, c2;

// main class
public static void main(String[] args)
{
    // create a new frame
    f = new JFrame("frame");

    // create a object
    solve s = new solve();

    // array of string containing cities
    String s1[] = { "Jalpaiguri", "Mumbai", "Noida", "Kolkata", "New Delhi" };
    String s2[] = { "male", "female", "others" };

    // create checkbox
    c1 = new JComboBox(s1);
    c2 = new JComboBox(s2);

    // set Kolkata and male as selected items
    // using setSelectedIndex
    c1.setSelectedIndex(3);
    c2.setSelectedIndex(0);

    // add ItemListener
    c1.addItemListener(s);
    c2.addItemListener(s);

    // set the checkbox as editable
    c1.setEditable(true);

    // create labels
    l = new JLabel("select your city ");
    l1 = new JLabel("Jalpaiguri selected");
    l3 = new JLabel("select your gender ");
    l4 = new JLabel("Male selected");

    // set color of text
    l.setForeground(Color.red);
    l1.setForeground(Color.blue);
    l3.setForeground(Color.red);
    l4.setForeground(Color.blue);
}
```

```
// create a new panel
JPanel p = new JPanel();

p.add(l1);

// add combobox to panel
p.add(c1);

p.add(l1);

p.add(l3);

// add combobox to panel
p.add(c2);

p.add(l4);

// set a layout for panel
p.setLayout(new FlowLayout());

// add panel to frame
f.add(p);

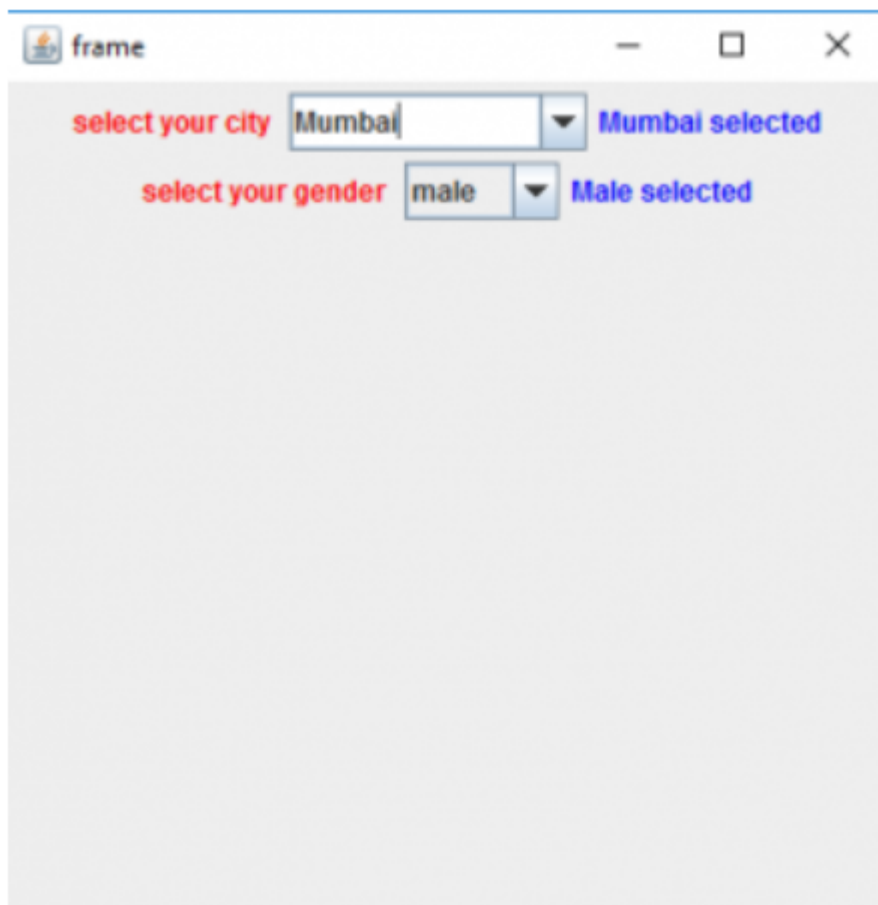
// set the size of frame
f.setSize(400, 400);

f.show();
}
public void itemStateChanged(ItemEvent e)
{
    // if the state combobox 1 is changed
    if (e.getSource() == c1) {

        l1.setText(c1.getSelectedItem() + " selected");
    }

    // if state of combobox 2 is changed
    else
        l4.setText(c2.getSelectedItem() + " selected");
}
}
```

**Salidad:**



## Bibliografía:

- Extraído de [Java Swing - JComboBox con ejemplos](#). Consultado el 28 de noviembre del 2022