

JTextField

JTextField es una parte del paquete javax.swing. La clase JTextField es un componente que permite la edición de una sola línea de texto. JTextField hereda la clase JTextComponent y utiliza la interfaz SwingConstants.

Los constructores de la clase son:

1. **JTextField()** : constructor que crea un nuevo TextField
2. **JTextField(int columns)** : constructor que crea un nuevo TextField vacío con un número especificado de columnas.
3. **JTextField(String text)** : constructor que crea un nuevo campo de texto vacío inicializado con la cadena dada.
4. **JTextField(String text, int columns)** : constructor que crea un nuevo textField vacío con la cadena dada y un número especificado de columnas.
5. **JTextField(Document doc, String text, int columns)** : constructor que crea un textfield que utiliza el modelo de almacenamiento de texto dado y el número dado de columnas.

Los métodos del JTextField son:

1. **setColumns(int n)** :establece el número de columnas del campo de texto.
2. **setFont(Font f)** : establece la fuente del texto que se muestra en el campo de texto.
3. **addActionListener(ActionListener l)** : establece un ActionListener en el campo de texto.
4. **int getColumns()** :get el número de columnas en el campo de texto.

Los siguientes son los programas para implementar JTextField.

1. Programa para crear un campo de texto en blanco de número definido de columnas.

```
// Java program to create a blank text
// field of definite number of columns.
import java.awt.event.*;
import javax.swing.*;
class text extends JFrame implements ActionListener {
    // JTextField
    static JTextField t;

    // JFrame
    static JFrame f;

    // JButton
    static JButton b;

    // label to display text
    static JLabel l;

    // default constructor
```

```
text()
{
}

// main class
public static void main(String[] args)
{
    // create a new frame to store text field and button
    f = new JFrame("textfield");

    // create a label to display text
    l = new JLabel("nothing entered");

    // create a new button
    b = new JButton("submit");

    // create a object of the text class
    text te = new text();

    // addActionListener to button
    b.addActionListener(te);

    // create a object of JTextField with 16 columns
    t = new JTextField(16);

    // create a panel to add buttons and textfield
    JPanel p = new JPanel();

    // add buttons and textfield to panel
    p.add(t);
    p.add(b);
    p.add(l);

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

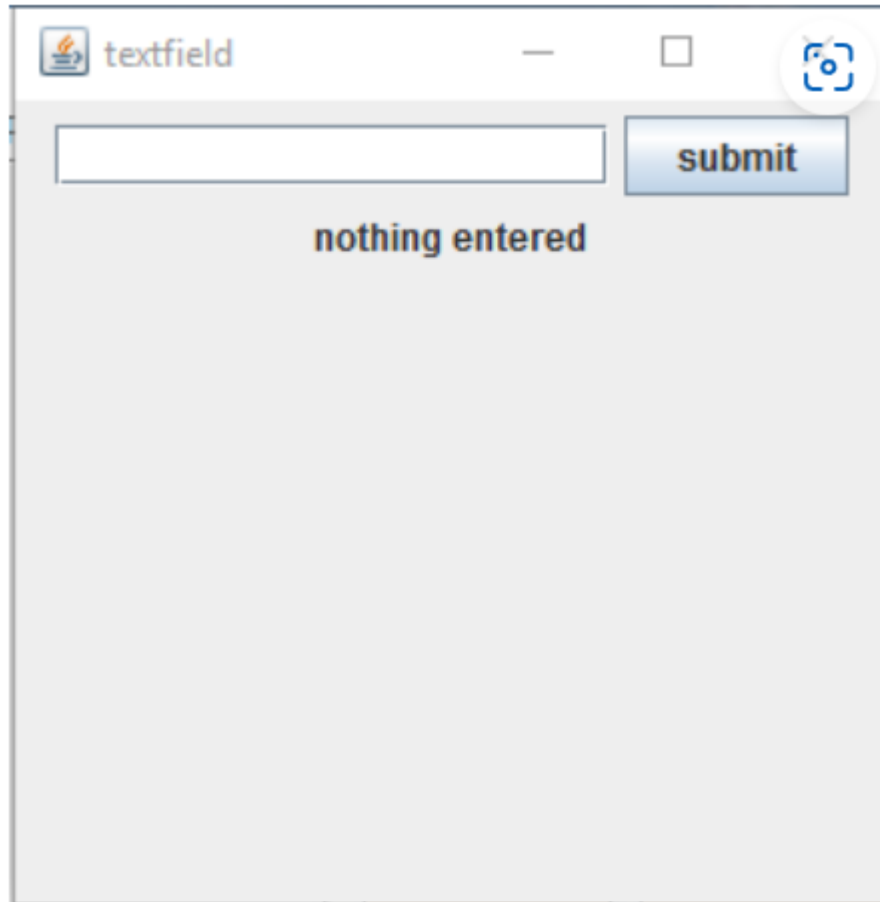
    // set the size of frame
    f.setSize(300, 300);

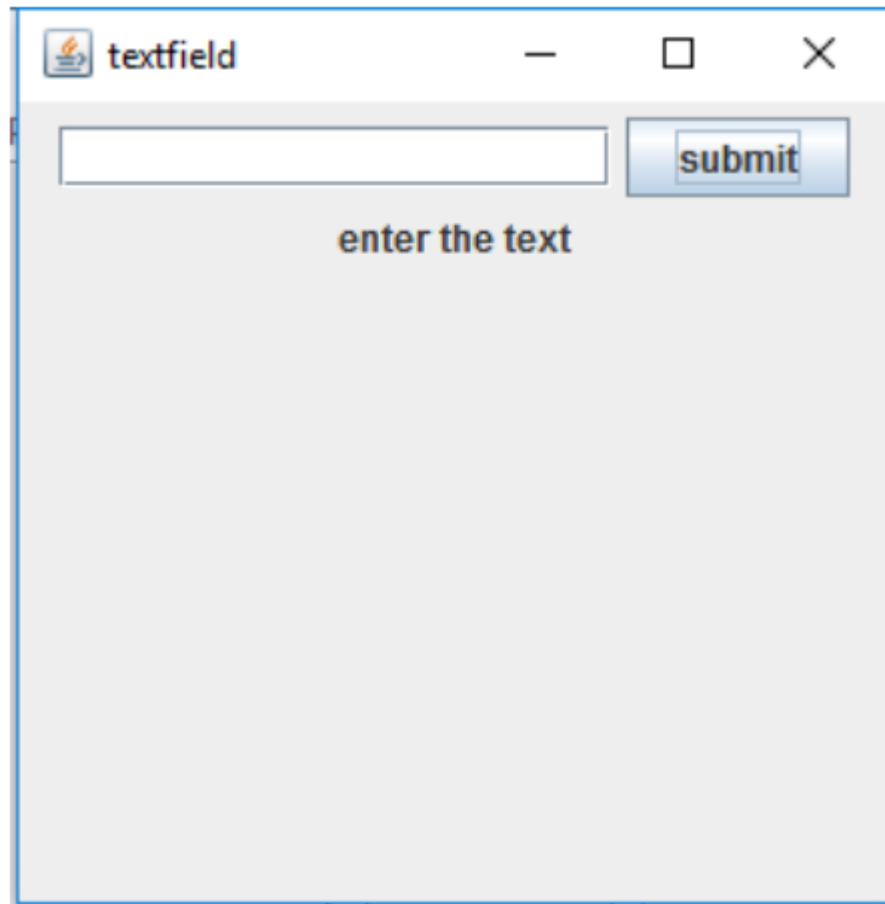
    f.show();
}

// if the button is pressed
public void actionPerformed(ActionEvent e)
{
    String s = e.getActionCommand();
    if (s.equals("submit")) {
```

```
// set the text of the label to the text of the field  
l.setText(t.getText());  
  
// set the text of field to blank  
t.setText(" ");  
}  
}  
}
```

salida:





[DescargarEjemplo](#)

2. Programa para crear un campo de texto en blanco con un texto inicial dado y un número dado de columnas

```
// Java program to create a blank text field with a
// given initial text and given number of columns
import java.awt.event.*;
import javax.swing.*;
class text extends JFrame implements ActionListener {
    // JTextField
    static JTextField t;

    // JFrame
    static JFrame f;

    // JButton
    static JButton b;

    // label to display text
    static JLabel l;

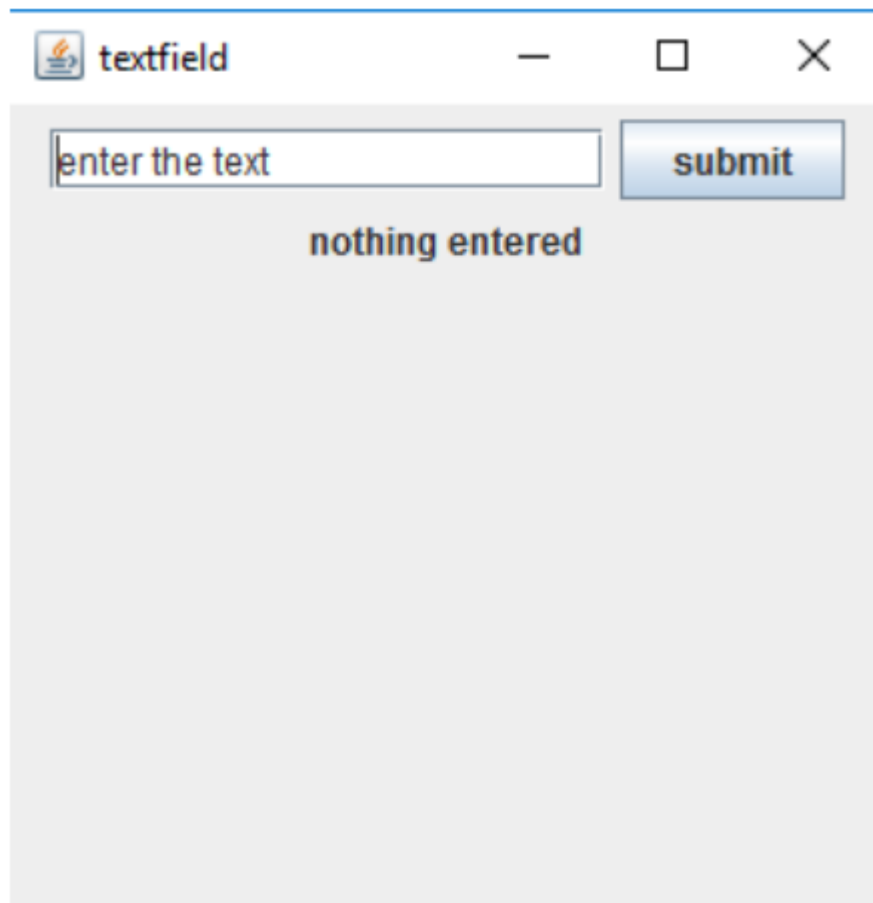
    // default constructor
    text()
```

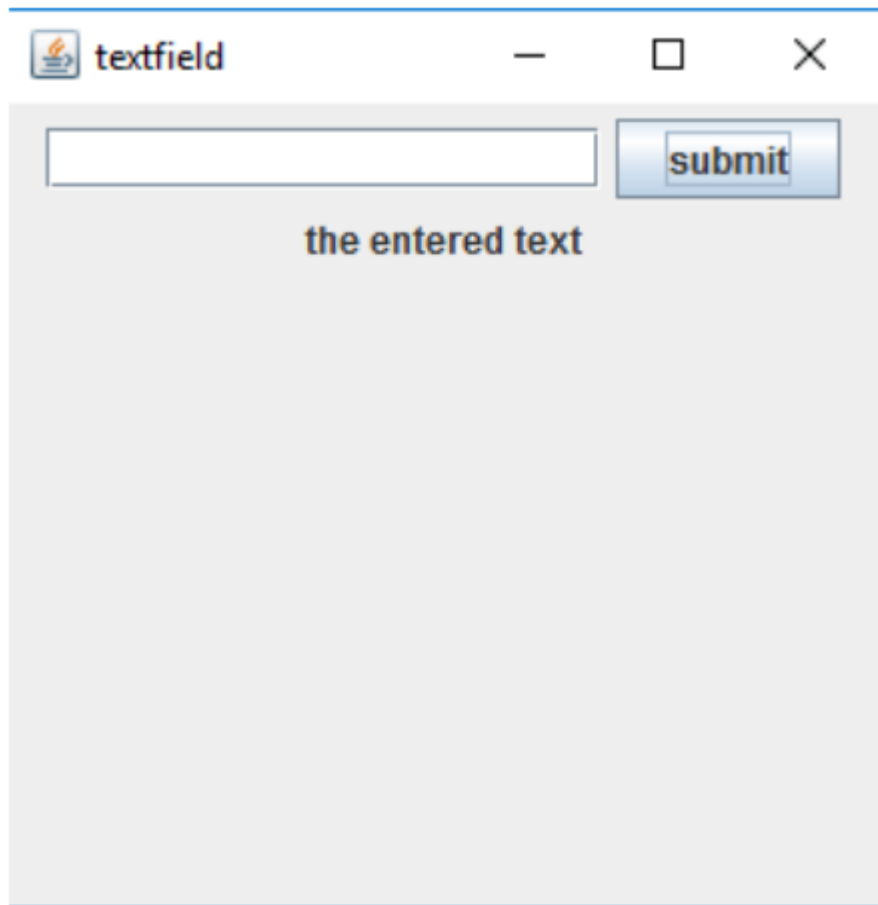
```
{  
}  
  
// main class  
public static void main(String[] args)  
{  
    // create a new frame to store text field and button  
    f = new JFrame("textfield");  
  
    // create a label to display text  
    l = new JLabel("nothing entered");  
  
    // create a new button  
    b = new JButton("submit");  
  
    // create a object of the text class  
    text te = new text();  
  
    // addActionListener to button  
    b.addActionListener(te);  
  
    // create a object of JTextField with 16 columns and a given initial text  
    t = new JTextField("enter the text", 16);  
  
    // create a panel to add buttons and textfield  
    JPanel p = new JPanel();  
  
    // add buttons and textfield to panel  
    p.add(t);  
    p.add(b);  
    p.add(l);  
  
    // add panel to frame  
    f.add(p);  
  
    // set the size of frame  
    f.setSize(300, 300);  
  
    f.show();  
}  
  
// if the button is pressed  
public void actionPerformed(ActionEvent e)  
{  
    String s = e.getActionCommand();  
    if (s.equals("submit")) {  
        // set the text of the label to the text of the field
```

```
l.setText(t.getText());

// set the text of field to blank
t.setText(" ");
    }
}
}
```

salida:





[DescargarEjemplo](#)

3. Programa para crear un campo de texto en blanco y establecer el tipo de fuente BOLD

```
// Java program to create a blank text field and set BOLD font type
import java.awt.event.*;
import java.awt.*;
import javax.swing.*;
class text extends JFrame implements ActionListener {
    // JTextField
    static JTextField t;

    // JFrame
    static JFrame f;

    // JButton
    static JButton b;

    // label to display text
    static JLabel l;

    // default constructor
    text()
    {
```

```
}

// main class
public static void main(String[] args)
{
    // create a new frame to store text field and button
    f = new JFrame("textfield");

    // create a label to display text
    l = new JLabel("nothing entered");

    // create a new button
    b = new JButton("submit");

    // create a object of the text class
    text te = new text();

    // addActionListener to button
    b.addActionListener(te);

    // create a object of JTextField with 16 columns
    t = new JTextField(16);

    // create an object of font type
    Font fo = new Font("Serif", Font.BOLD, 20);

    // set the font of the textfield
    t.setFont(fo);

    // create a panel to add buttons and textfield
    JPanel p = new JPanel();

    // add buttons and textfield to panel
    p.add(t);
    p.add(b);
    p.add(l);

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

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

    f.show();
}

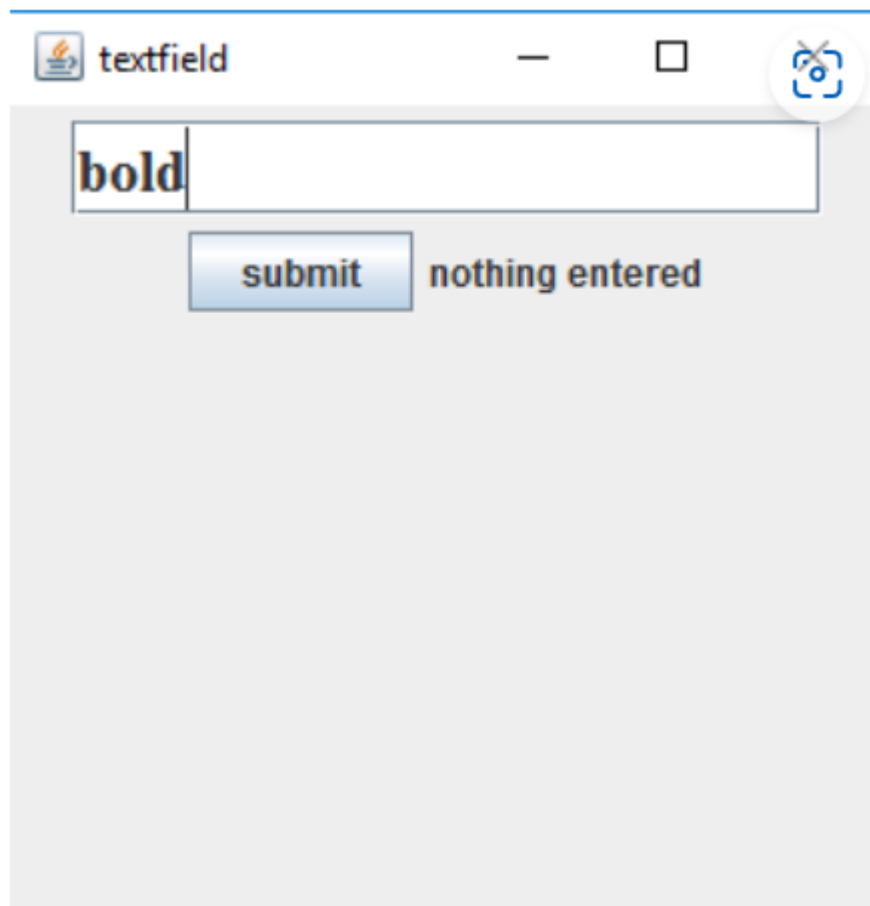
// if the button is pressed
```



```
public void actionPerformed(ActionEvent e)
{
    String s = e.getActionCommand();
    if (s.equals("submit")) {
        // set the text of the label to the text of the field
        l.setText(t.getText());

        // set the text of field to blank
        t.setText(" ");
    }
}
```

salida:



[DescargarEjemplo](#)

Bibliografía:

- Extraído de <<Java Swing | JTextField>>. Consultado el 28 de noviembre del 2022