

Leggere testo di input con applicazioni GUI e il package *swing*

Esempio di **applicazione** di tipo GUI che usa una **semplice finestra modale**:

```
import javax.swing.*;
import java.awt.*;

public class Finestre extends JFrame {

    String nome="";

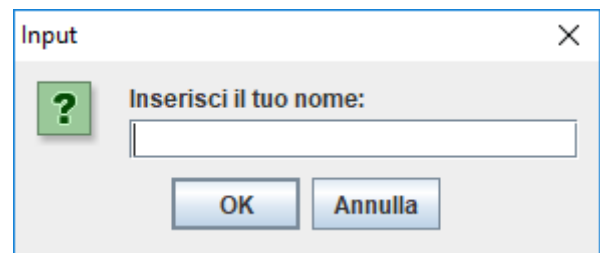
    public Finestre () {

        nome = JOptionPane.showInputDialog("Inserisci il tuo nome: ");

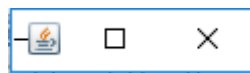
        System.out.println("Ciao " + nome); // visualizzazione a monitor

        pack(); // minimizzazione delle dimensioni della finestra parent
        setVisible(true);
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String [] args){
        new Finestre(); // finestra parent anonima
    }
}
```



nb: alla pressione di **Annulla**
si visualizza la finestra *parent*



General Output

```
-----
Ciao null
```

e, scrivendo a monitor, si evidenzia che invece della stringa attesa
il metodo ritorna **null**

Volendo evitare questo caso, si può inserire un semplice controllo:

```
if (nome !=null)
    System.out.println("Ciao " + nome);
else
    System.out.println("Ciao anonimo");
```

General Output

```
-----
Ciao anonimo
```

Lo stesso esempio realizzato come [Applet](#) che usa una **semplice finestra modale**:

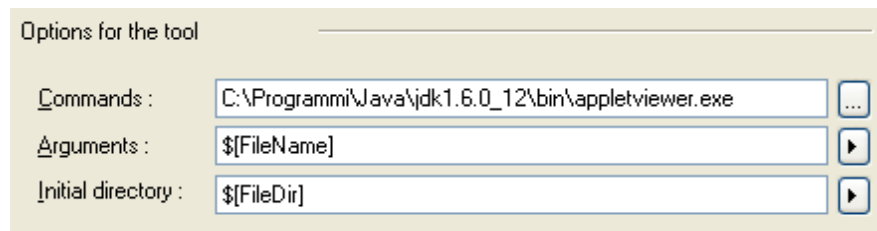
```
// da eseguire con appletviewer da linea di comando > appletviewer Finestre.html  
// oppure con opportune selezioni nel Plug-in (da Pannello di controllo)  
// oppure in ambiente JCreator
```

```
import javax.swing.*;  
import java.awt.*;  
import java.applet.*;  
  
public class Finestre extends JApplet {  
  
    String nome="";  
  
    public void init () {           // metodo init() per avviare  
  
        nome = JOptionPane.showInputDialog("Inserisci il tuo nome: ");  
    }  
  
    public void paint(Graphics g) {  
  
        g.drawString("Ciao " + nome, 10, 50);           // disegno della stringa  
    }  
}
```

Con codice HTML (file *Finestre.html*):

```
<html>  
  <head>  
    <title>Dialog box per input </title>  
  </head>  
  <body>  
    <object code ="Finestre.class" width = "300" height = "100">  
      </object>  
  </body>  
</html>
```

nb: per eseguire la pagina html con programma **appletviewer**, inserito come **tools** in JCreator, impostare come opzioni:



Per scoprire le potenzialità di swing¹ si poteva eseguire la **demo** in file compresso seguendo percorso

C:\Programmi\Java\jdk1.6.0_21\demo\jfc\SwingSet2\SwingSet2.jar

Si può consultare la [sezione](#) del [Java Tutorial](#) oggi disponibile nel sito della Oracle



¹ SWING <http://java.sun.com/docs/books/tutorial/uiswing/index.html>

Esempio senza apparente gestione di eventi

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class AppIcone extends JFrame {

    private JButton btnIcone;
    private JLabel testo;

    private Icon[] figure = { new ImageIcon ("../src/IMG/Inverno.gif"),
                             new ImageIcon ("../src/IMG/Estate.gif"),
                             new ImageIcon ("../src/IMG/Ninfee.gif") };

    // percorso relativo alla sottocartella classes in cui è memorizzato il bytecode
    // se nella stessa cartella ./nomeImage

    public AppIcone () {

        Container cnt = getContentPane(); // di default con gestore di layout tipo BorderLayout

        btnIcone = new JButton (figure[1]); // immagine se il mouse è al di fuori della finestra

        // di default setRolloverEnabled (true)
        // abilitazione del riconoscimento del trascinamento del mouse (roll over)

        btnIcone.setRolloverIcon (figure[0]); // definizione dell'icona da usare durante lo scorrimento del mouse
        btnIcone.setPressedIcon (figure[2]); // definizione dell'icona da usare a seguito di click sul mouse

        btnIcone.setToolTipText ("Spendido inverno! Premi per vedere ninfee! ");
        cnt.add (btnIcone, BorderLayout.CENTER);

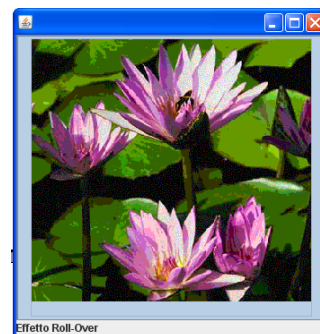
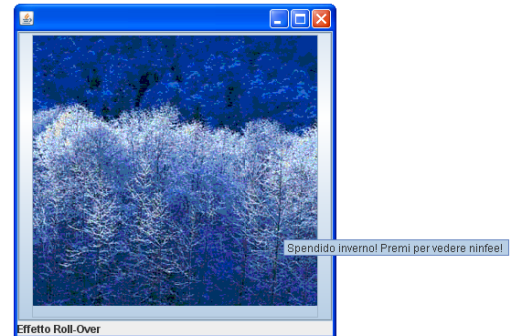
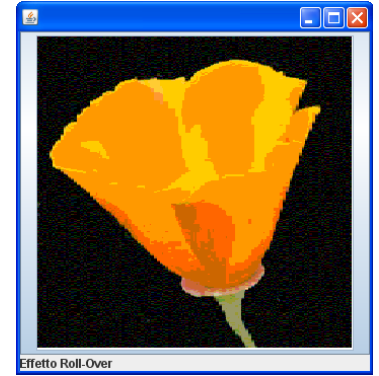
        testo = new JLabel("Effetto Roll-Over");
        cnt.add (testo, BorderLayout.PAGE_END);

        pack();
        setVisible (true);
        setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
    }

    public static void main(String args []){

        AppIcone o = new AppIcone();

    }
}
```



nb: salvando in stessa cartella il sorgente e la sottocartella che contiene le immagini **IMG**