

El control Lista

Seleccionar un elemento de la lista

En una lista existen dos posibles acciones, seleccionar un elemento de la lista cuando se sitúa el cursor sobre dicho elemento, o hacer doble-clic sobre un elemento de la lista.

Propósito

Un control lista tiene como elementos los nombres de los tres colores básicos: rojo, verde y azul. Al seleccionar un elemento de la lista se pinta un rectángulo del color seleccionado.

El control *List*

Para crear un control *lista* de la clase *List* y añadirle elementos, se procede del siguiente modo

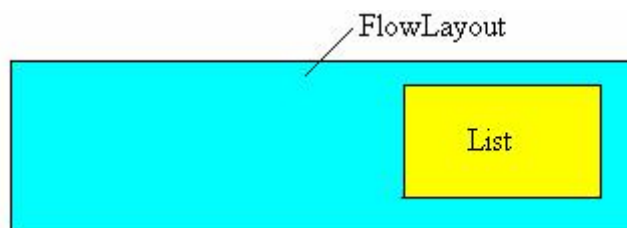
```
List lista=new List();  
lista.add("Rojo");  
lista.add("Verde");  
lista.add("Azul");  
lista.select(0);
```

La última línea de código, indica que el primer elemento de la lista, "Rojo", aparecerá inicialmente seleccionado.

Para saber qué elemento de la lista ha sido seleccionado empleamos dos funciones *getSelectedIndex* y *getSelectedItem*, el primero devuelve un entero que indica la posición del elemento en la lista, y la segunda devuelve un string que guarda el elemento de la lista.

```
int indice=lista.getSelectedIndex();  
string texto=lista.getSelectedItem();
```

Diseño



Crear un applet y situar sobre el applet en el modo diseño un control lista (*List*)

Establecer *FlowLayout* como gestor de diseño del applet, con alineamiento a la derecha (la propiedad **alignment** vale 2).

Crear un array de objetos de la clase *Color* cuyos elementos son los tres colores básicos

```
Color[] colores={Color.red, Color.green, Color.blue};
```

Redefinir la función *paint* para pintar un rectángulo (a la izquierda del applet) del color seleccionado

Respuesta a las acciones del usuario

La respuesta a la acción de seleccionar un elemento de la lista la podemos resumir en los siguientes puntos:

- Se crea una clase que implemente el interface *ItemListener*, y que ha de definir *itemStateChanged*

```
class ItemLista implements ItemListener{
    private ListaApplet2 applet;
    public ItemLista(ListaApplet2 applet){
        this.applet=applet;
    }
    public void itemStateChanged(ItemEvent ev){
        applet.funcionRespuesta (ev);
    }
}
```

- Mediante *addItemListener* se conecta el componente con un objeto de la clase que maneja los sucesos originados en dicho componente.

```
ItemLista accion=new ItemLista(this);
lista.addItemListener(accion);
```

- La información acerca del suceso viene encapsulada en un objeto de la clase *ItemEvent*.

La definición de la función respuesta es la misma que en el ejemplo anterior

```
public class ListaApplet2 extends Applet {
    List lista=new List();
    FlowLayout flowLayout1 = new FlowLayout();
    final Color[] colores={Color.red, Color.green, Color.blue};
    int indice;

    public void init() {
        flowLayout1.setAlignment(2);
        ItemLista accion=new ItemLista(this);
        lista.addItemListener(accion);
        lista.add("Rojo");
        lista.add("Verde");
        lista.add("Azul");
        lista.select(0);
        this.setLayout(flowLayout1);
    }
}
```

```

        this.add(lista, null);
    }
    void funcionRespuesta(ItemEvent e) {
        indice=lista.getSelectedIndex();
        repaint();
    }

    public void paint(Graphics g){
        g.setColor(colores[indice]);
        g.fillRect(2, 2, 100, 50);
    }
}

//*****
class ItemLista implements ItemListener{
    private ListaApplet2 applet;
    public ItemLista(ListaApplet2 applet){
        this.applet=applet;
    }
    public void itemStateChanged(ItemEvent ev){
        applet.funcionRespuesta (ev);
    }
}

```

El control selección (*Choice*)

El uso del control selección (*Choice*) es similar al uso del control lista (*List*), la diferencia está en el espacio que ocupa en el applet. El control *Choice* se puede desplegar pulsando en la flecha situada en la parte derecha para mostrar los elementos que contiene y seleccionar uno de ellos, que aparece a la izquierda de la flecha de la caja combinada. El control *Choice* implementa el interface *ItemListener* de la misma manera que el control *List*.

Para crear un control *lista* de la clase *Choice* y añadirle elementos, se procede del siguiente modo

```

Choice lista=new Choice();
lista.add("Rojo");
lista.add("Verde");
lista.add("Azul");
lista.select(1);

```

La última línea de código, indica que el segundo elemento de la lista, "Verde", aparecerá inicialmente seleccionado a la izquierda de la flecha en la caja combinada.