Contains broken english... :-)

WidgetServer tutorial.

WidgetServer can make windows. In this work, it uses a very easy language. Programs can deal with WidgetServer using standard input and they can get widgets events in standar output.

WidgetServer is independent of programming languaje. It can create windows from C, Python, Java, Perl, Bash,...

Fast example.

This code represents a window in WidgetServer:

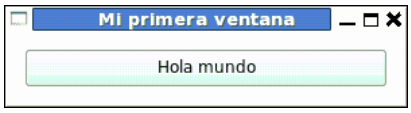
```
<window:w1>
      <title:Mi primera ventana/>
      <button:b1>
            <text>
                  Hola mundo
            </text>
      </button>
</window>
```

Important: Every tag must be written in a new line. You must left one blank line at end.

This window can be created by WidgetServer. You must write them in WidgetServer standar input or save them in fila, per example, *ej1.ws* and write this shell command:

```
cat ej1.ws|widgetserver
```

Output is:



Example before shows a window with button, but you can't read events from button.

If you want to read those events:

```
<window:w1>
      <title:Mi primera ventana/>
      <button:b1>
            <text>
                  Hola mundo
            </text>
            <listen:clicked/>
      </button>
</window>
```

New command <listen:clicked/> makes WidgetServer writes mouse click button events. Putting this code in file ej2.ws and writing:

```
cat ej2.ws|widgetserver
```

You will obtain a window with a button. Click over button generates:

```
*clicked: b1
```

This says that b1 has received a click event.

Events writes mensajes like:

**Event name*: *widget name*

Tags and Widgets.

In html every element like <text> is called tag. WidgetServer tags are similar as html tags.

Every object shows by WidgetServer is called Widget. Widget are represented by a pair of tags:

```
<widget type:widget name>
      ...
</widget type>
```

Between that tags widget properties are written. Per example:

```
<window:w1>
      <title:Mi first window/>
</window>
```

It says that new widget will be make. This widget will be a window. Window name will be “Mi first window”.

Tags without properties are written like:

```
<texto del tag/>
```

Pex example tag title from widget window.

Command line.

WidgetServer is executed using command *widgetserver*. –help option shows:

```
widgetserver [--help] [-debug] [-exec path]
Starts WidgetServer
--help Shows this help
-debug shows debug output
-exec path Executes path in WidgetServer. All input and output are redirected by
WidgetServer
```

--exec path option is very usefull for programming. Process only sends their output and reads input using standart input/output. WidgetServer reads and writes using standar output/input of process.

Exec widgetserver without options: WidgetServer waits to new commands reading its standart input.

Widgets types.

We will study widgets.

Widget: Label.

The Label widget provides a text or html display.

Syntax:

```
<label:widget name>
      ...text or html...
</label>
```

Example:

```
<window:w1>
      <label:l1>
            Your text
```

```
      </label>
      <label:l2>
            <html>
                  <b>Yes, is HTML!!</b>
            </html>
      </label>
</window>
```

Html can be written using <html> </html> tags. The window from this example is:

Widget: Window.

This widget shows windows. Syntax:

```
<window: widget name>
Widgets and properties
</window>
```

Properties are:

```
<title: Title/>
```

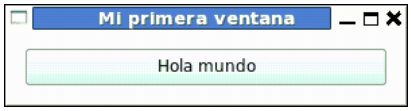
Window title.

```
<icon:path/>
```

Path for window icon.

Widget: Button.

The Button widget provides a command button.



Syntax:

```
<button:widget name>
      ...Properties...
</button>
```

Properties:

```
<text>
      ...text...
</text>
```

Sets widget text. Blank spaces are trimmed.

```
<text end="End text">
      ...text...
</text end="End text">
```

Sets widget text. Blanck spaces are not trimmed.

```
<icon:path/>
```

Path for window icon.

```
<listen:clicked/>
```

Requires mouse click event.

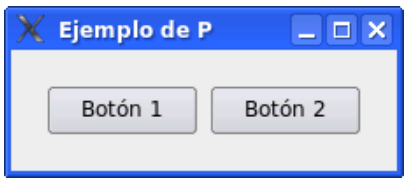
Icon and listen are not needed.

Example:

```
<window:w1>
	<title:Mi primera ventana/>
	<button:b1>
		<text>
			Hola mundo
		</text>
		<listen:clicked/>
		<icon:bookmark.png/>
	</button>
</window>
```

Widget: P.

The P widget lines up widgets horizontally.



Syntax:

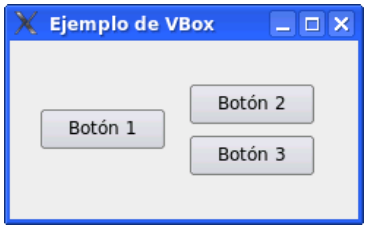
```
<p>
...Widgets...
</p>
```

Example:

```
<window:w1>
	<title:Ejemplo de P/>
	<p>
		<button:b1>
			<text>
				Botón 1
			</text>
		</button>
		<button:b2>
			<text>
				Botón 2
			</text>
		</button>
	</p>
</window>
```

Widget: VBox.

The VBox widget lines up widgets vertically.



Syntax:

```
<vbox>
...Widgets...
```

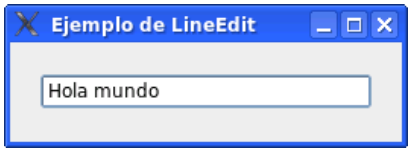
```
</vbox>
```

Example:

```
<window:w1>
	<title:Ejemplo de VBox/>
	<p>
		<button:b1>
			<text>
				Botón 1
			</text>
		</button>
		<vbox>
			<button:b2>
				<text>
					Botón 2
				</text>
			</button>
			<button:b3>
				<text>
					Botón 3
				</text>
			</button>
		</vbox>
	</p>
</window>
```

Widget: LineEdit.

The LineEdit widget is a one-line text editor.



Syntax:

```
<lineedit: widget name>
	...Properties...
</lineedit>
```

Properties are:

```
<text: Text />
```

This property holds the line edit's text.

```
<getText/>
```

gets text from widget.

Example:

```
<window:w1>
	<title:Ejemplo de LineEdit/>
	<p>
		<lineedit:le1>
			<text:Hola mundo/>
		</lineedit>
	</p>
</window>
```

Widget: TextEdit.

The TextEdit widget provides a widget that is used to edit and display text.

Syntax:

```
<textedit: widget name>
      ...Properties...
</textedit>
```

Properties:

```
<text>
      ...text...
</text>
```

Sets widget text. Blanck spaces are trimmed.

```
<text end="End text">
      ...text...
</text end="End text">
```

Sets widget text. Blanck spaces are not trimmed.

```
<getText/>
```

gets text from widget. First it writes text's number of bytes. Second text is written.

Example:

```
<window:w1>
      <title:Ejemplo de TextEdit/>
      <p>
            <textedit:le1>
                  <text>
                        Hola mundo.
                  </text>
            </textedit>
      </p>
</window>
```

Widget: HTML.

The HTML widget provides a rich text browser with hypertext navigation.

Syntax:

```
<html: widget name>
...html...
</html>
```

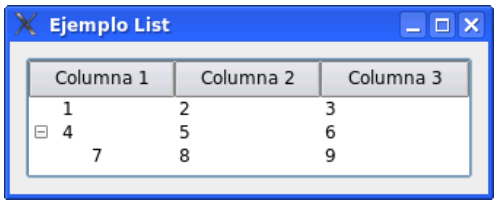
This html is very simple. Html code can be written in the same line.

Example:

```
<window:w1>
	<title:Ejemplo de Html/>
	<html:html1>
		<b>Esto es un ejemplo de html.</b><br>
		<center><img src="bookmark.png"></center>
	</html>
</window>
```

Widget: List.

The List widget provides a tree/list view of text.



Syntax:

```
<list: widget name>
	...Properties...
	..items...
</list>
```

Properties

```
<selection mode:multiple/>
```

Several rows can be selected.

```
<selection mode:simple/>
```

Only one row can be selected.

```
<listen:selection changed/>
```

Listen selection events.

Items are text rows:

```
<item:item name>
	...Data rows. One line for column...
	...Others items...
</item>
```

Headers:

```
<headers>
	...Header's name. One line for column...
</headers>
```

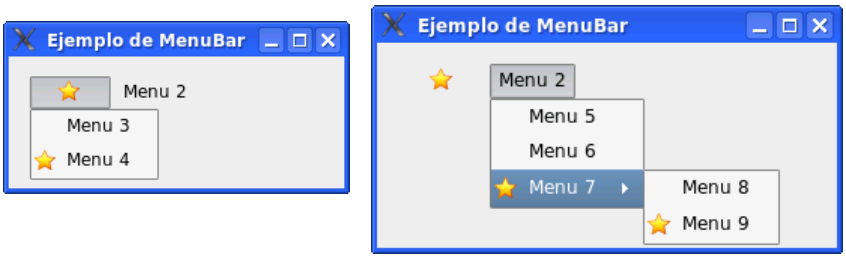
Example:

```
<window:w1>
	<title:Ejemplo List/>
	<list:l1>
		<listen:selection changed/>
		<selection mode:multiple/>
		<headers>
			Columna 1
			Columna 2
			Columna 3
		</headers>
		<item:i1>
			1
			2
			3
```

```
            </item>
            <item:i2>
                  4
                  5
                  6
                  <item:i3>
                        7
                        8
                        9
                  </item>
            </item>
      </list>
</window>
```

Widget: MenuBar.

The MenuBar widget provides a horizontal menu bar.



Syntax:

```
<menubar: widget name>
      ..items...
</menubar>
```

Items are menus. There are two types of items:

- Items without childs:

  Syntax:

  ```
  <item:item_name text=”Menu text” icon=”path”/>
  ```

  Icon property is not needed.

- Items with childs:

  ```
  <item:item_name text=”Menu text” icon=”path”>
        ...items...
  </item>
  ```

  Icon property is not needed.

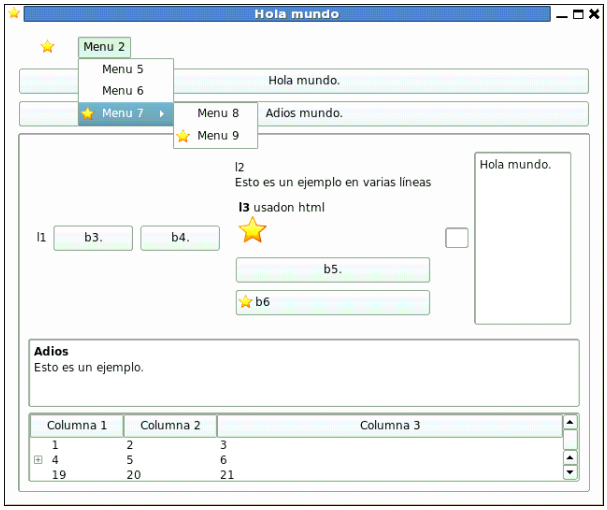
Example:

```
<window:w1>
      <title:Ejemplo de MenuBar/>
      <menubar:menubar1>
            <item:m1 text="Menu 1" icon="bookmark.png">
                  <item:m3 text="Menu 3"/>
                  <item:m4 text="Menu 4" icon="bookmark.png"/>
            </item>
            <item:m2 text="Menu 2">
                  <item:m5 text="Menu 5"/>
                  <item:m6 text="Menu 6"/>
                  <item:m7 text="Menu 7" icon="bookmark.png">
                        <item:m8 text="Menu 8"/>
                        <item:m9 text="Menu 9" icon="bookmark.png"/>
                  </item>
            </item>
```

```
	</menubar>
</window>
```

Full example.

We are going to see code for this window:



Code is:

```
<window:w>
	<title: Hola mundo/>
	<icon:bookmark.png/>
	<menubar:menubar1>
		<item:m1 text="Menu 1" icon="bookmark.png">
			<item:m3 text="Menu 3"/>
			<item:m4 text="Menu 4" icon="bookmark.png"/>
		</item>
		<item:m2 text="Menu 2">
			<item:m5 text="Menu 5"/>
			<item:m6 text="Menu 6"/>
			<item:m7 text="Menu 7" icon="bookmark.png">
				<item:m8 text="Menu 8"/>
				<item:m9 text="Menu 9" icon="bookmark.png"/>
			</item>
		</item>
	</menubar>
	<button:b1>
		<text>
		Hola mundo.
		</text>
	</button>
	<button:b2>
		<text>
		Adios mundo.
		</text>
	</button>
	<window:scroll>
	<p>
		<label:l1>
			l1
		</label>
		<button:b3>
			<text>
			b3.
```

```
            </text>
      </button>
      <button:b4>
            <text>
            b4.
            </text>
      </button>
      <vbox>
            <label:l2>
                  l2
                  Esto es un ejemplo en varias líneas
            </label>
            <label:l3>
                  <html>
                  <b>l3</b> usadon html<br>
                  <img src="bookmark.png">
                  </html>
            </label>
            <button:b5>
                  <text>
                  b5.
                  </text>
            </button>
            <button:b6>
                  <text>
                  b6
                  </text>
                  <icon:bookmark.png/>
                  <listen:clicked/>
            </button>
      </vbox>
      <lineedit:le1>
      <text:Hola mundo/>
      </lineedit>
      <textedit:te1>
            <text>
                  Hola mundo.
            </text>
      </textedit>
</p>
<html:html1>
      <b>Hola mundo</b><br>
      Esto es un ejemplo.
</html>
<list:list1>
      <listen:selection changed/>
      <selection mode:multiple/>
      <headers>
            Columna 1
            Columna 2
            Columna 3
      </headers>
      <item:pepe>
            1
            2
            3
      </item>
      <item>
            4
            5
            6
            <item>
                  7
                  8
```

```
                  9
            </item>
            <item>
                  10
                  11
                  12
            </item>
            <item>
                  13
                  14
                  15
                  <item>
                        16
                        17
                        18
                  </item>
            </item>
      </item>
      <item>
            19
            20
            21
      </item>
</list>
</window>
</window>

<lineedit:le1>
      <getText/>
</lineedit>

<html:html1>
      <b>Adios</b><br>
      Esto es un ejemplo.
</html>
```