

WIDGETS

Input Textbox

import android.widget.

Button mbutton;

mbutton = (Button) findViewById(R.id.button1);

TextView mtext;

mtext = (TextView) findViewById(R.id.numbered);

mbutton.setOnClickListener(new

New View.OnClickListener()

{ public void onClick(View v) {

});

editTexte = (EditText) findViewById(R.id.numbered);

checkbox

CHECK BOX

- Descende de TextView;

- Listener:

checkbox = ((CheckBox) findViewById(R.id.c))

b.setOnCheckedChangeListener(new CheckOnCheckedChangeListener() {

public void onCheckedChange(CompoundButton button, boolean b)

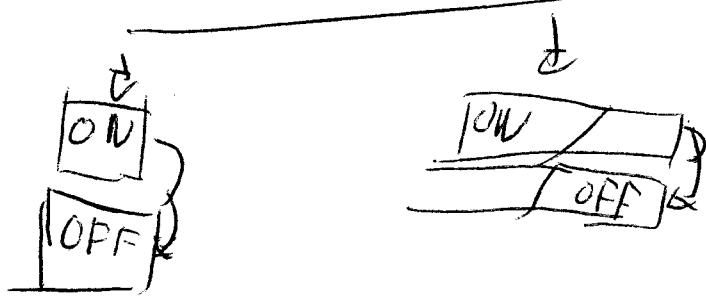
setVisibility → V. setVisibility (View.VISIBLE)

↳ R. setVisibility (View.INVISIBLE)

setVisibility → V. setVisibility (View.GONE)

↳ V. setVisibility (View.INVISIBLE)

TOGGLE BUTTON & SWITCH



android: checked = "true" or "false"

android: textOn = "ON".

android: textOff = "OFF".

Listener: equal to Radio Button & Checkbox.

IMAGEVIEW & IMAGEBUTTON

MYGIC VOOR IMAGEBUTTON

- equal to TextView > button can images.

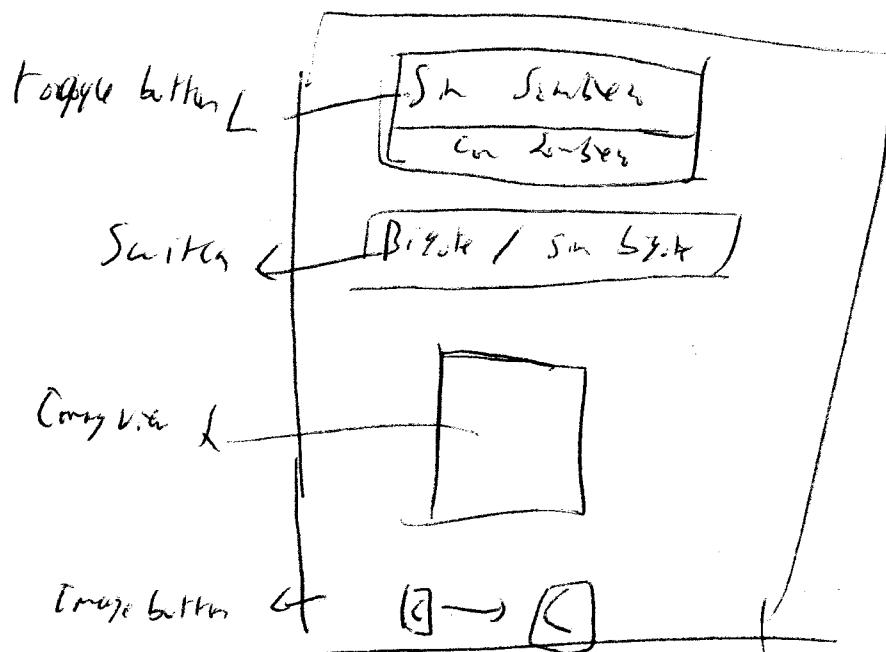
- can images be saved as Drawable.

- Format: PNG.

- (drawable) -> R.Drawable (drawable)

- property -> android:src.

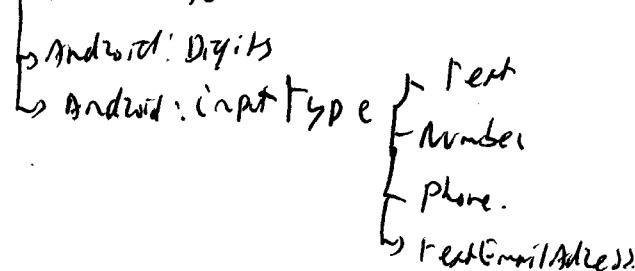
- Gehen Android Button direkt Carpet by new image subset.



EDIT TEXT

- Subclase de TextView.

• Propiedades: `android:autofillText`.



STRINGS.XML

Hay que mantener el texto fuera del código fuente.

. Puta `Res/values/strings.xml`.

Uso:

```

<resources>
  <string name="Residuo">Nube</string>
</resources>
  
```

name: Identifica el recurso.

~~Estoy~~

Etiquta String: Almacena el texto que describe el resultado.

referencia: Desde el Diccionario String / Result.

Desde el código: String s = getString(R.String.Result).

EJERCICIO PRACTICO

Re / Values → Lenguaje de salida

introduce Tu datos:

Nombre:

DNI:

Teléfono:

Email

Re / values - FR

Re / values - EN

Boton eliminar en Re
se pulsa en New Reference
selecciona la reg. en

el resultado.

(CONTENEDORES)

Gestionar la distribución de los widgets

Típos:

- RelativeLayout: Muestra widgets en reglas.
- LinearLayout: Muestra widgets en filas.
- TableLayout: Muestra widgets en celdas.

~~se p~~

Se pueden anidar uno dentro de otro.

RELATIVE LAYOUT

Permite a los hijos (contenidos) especificar su posición

con relación al padre (contenedor) o hermanos.

Propiedades:

Android: Layout_AlignParentTop = "top"	Le alinea a la parte superior del padre.
Android: Layout_CenterVertical = "true"	True o falso.
Android: Layout_CenterHorizontal = "true"	
Android: Layout_Below = "(id)"	Esto es, elemento debajo de "
Android: Layout_toRightOf = "(id)"	" "

LÍNEA/COLUMNAS

Alinear

permite alinear los controles horizontalmente en una fila o verticalmente en una columna.

Propiedades:

Android: orientation = {
 D "Vertical"
 A "Horizontal"

Android: layout_width = "wrap" "Wrap_Content"

Android: layout_height = "wrap_Content".

Android: layout_gravity = " " . ~~Este se alinea de ultimo~~

Todos se alinea por de arriba a abajo > de izquierda a derecha ~~horizontal~~
 por defecto.

TABLE (GRIDVIEW) > GRID LAYOUT

permite crear una distribución basada en filas y columnas, o, decir, mediante tablas.

Propiedades:

Android: RowCount = num } Especifica

Android: ColumnCount = num } numero de filas > columnas. Especifica el nro.

Android: Layout_Row = "Num". } Fijo que especificar en cada widget

Android: Layout_Column = "Num". } o que filas > columnas pertenezca

Para que un elemento ocupe más de una celda:

Android: Layout_ColumnSpan = "numero". Toma el espacio X columnas

Android: Layout_RowSpan = "numero". Toma el espacio X filas.

Android: Layout_Gravity = "fill" . Si fija para siempre.