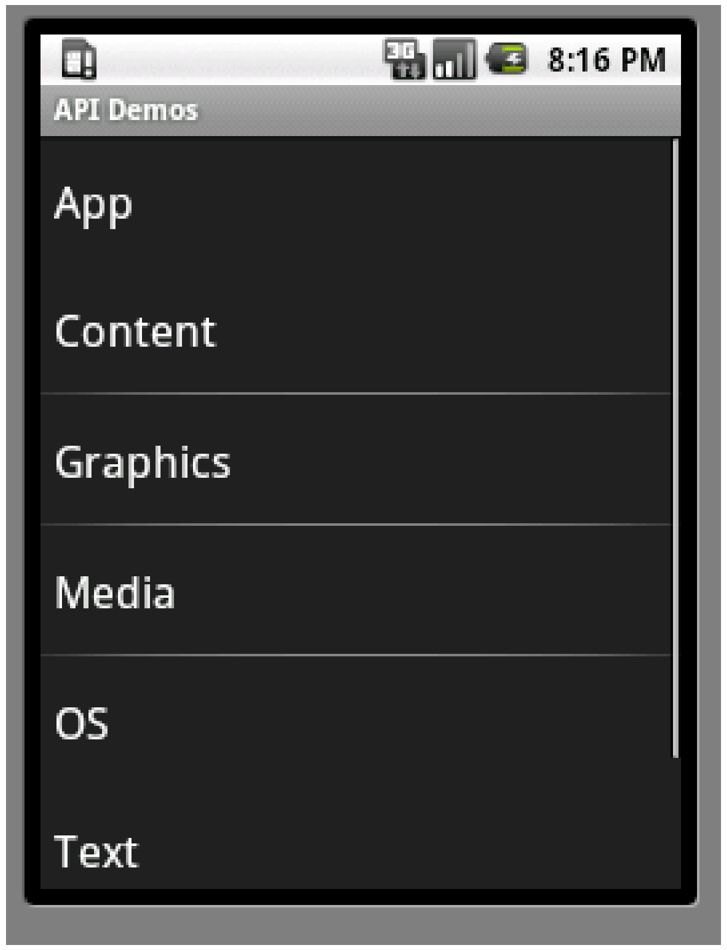
CS 696 Mobile Phone Application Development
Fall Semester, 2009
Doc 7 Attributes & Dialogs
Sept 22, 2009

API Demos


8:16 PM

API Demos

App

Content

Graphics

Media

OS

Text

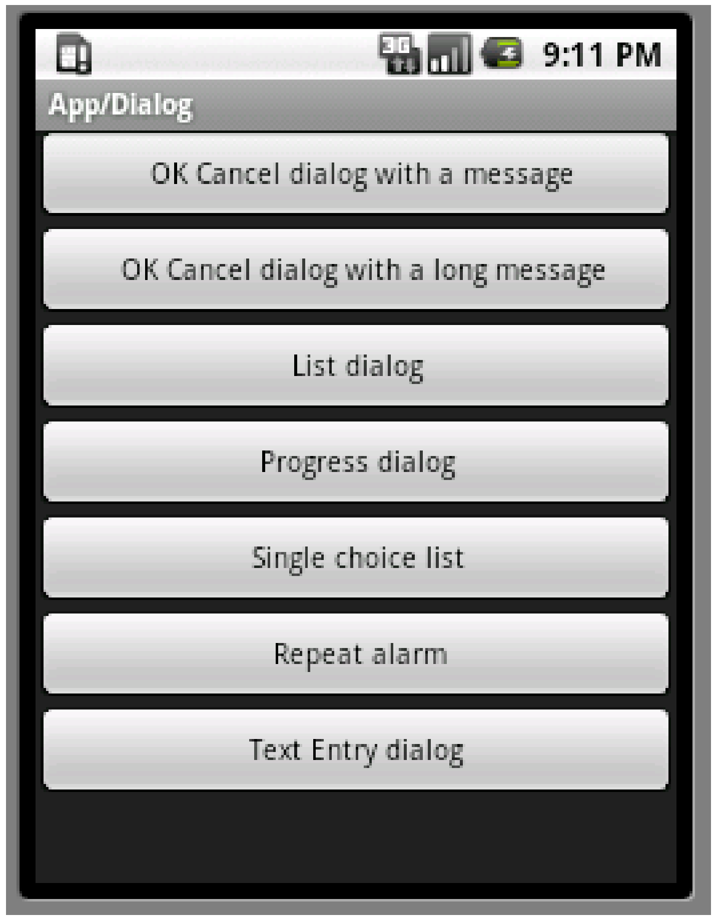
Source

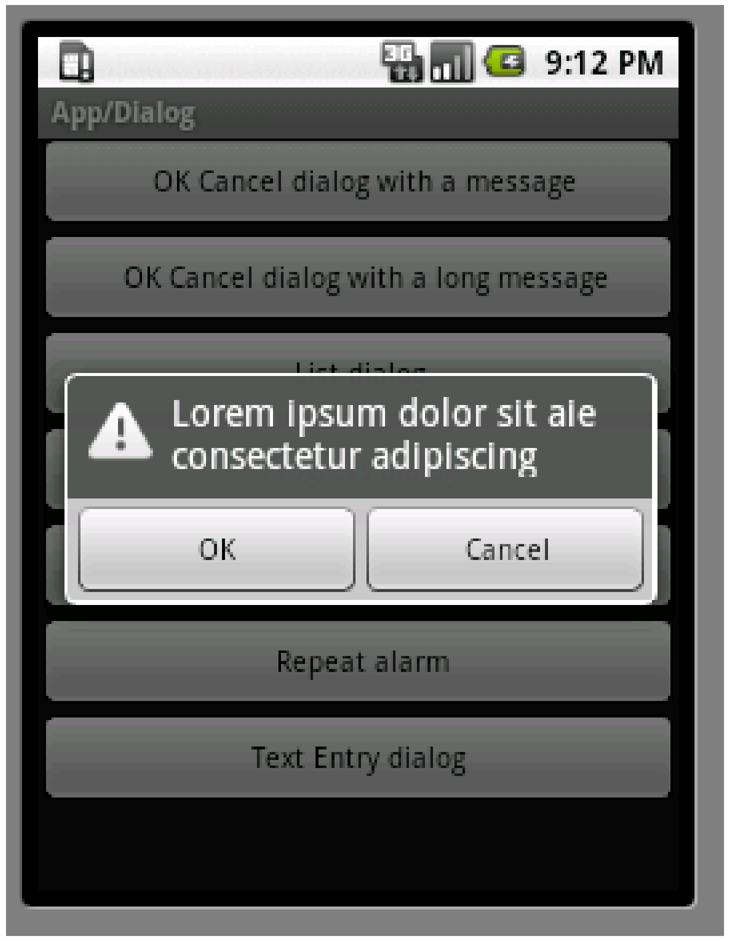
androidInstallation/platforms/android-1.6/samples/ApiDemos

On-line

http://developer.android.com/guide/samples/index.html

Dialog Demos




9:12 PM
App/Dialog
OK Cancel dialog with a message
OK Cancel dialog with a long message
Lorem ipsum dolor sit aie consectetur adipiscing
OK
Cancel
Repeat alarm
Text Entry dialog

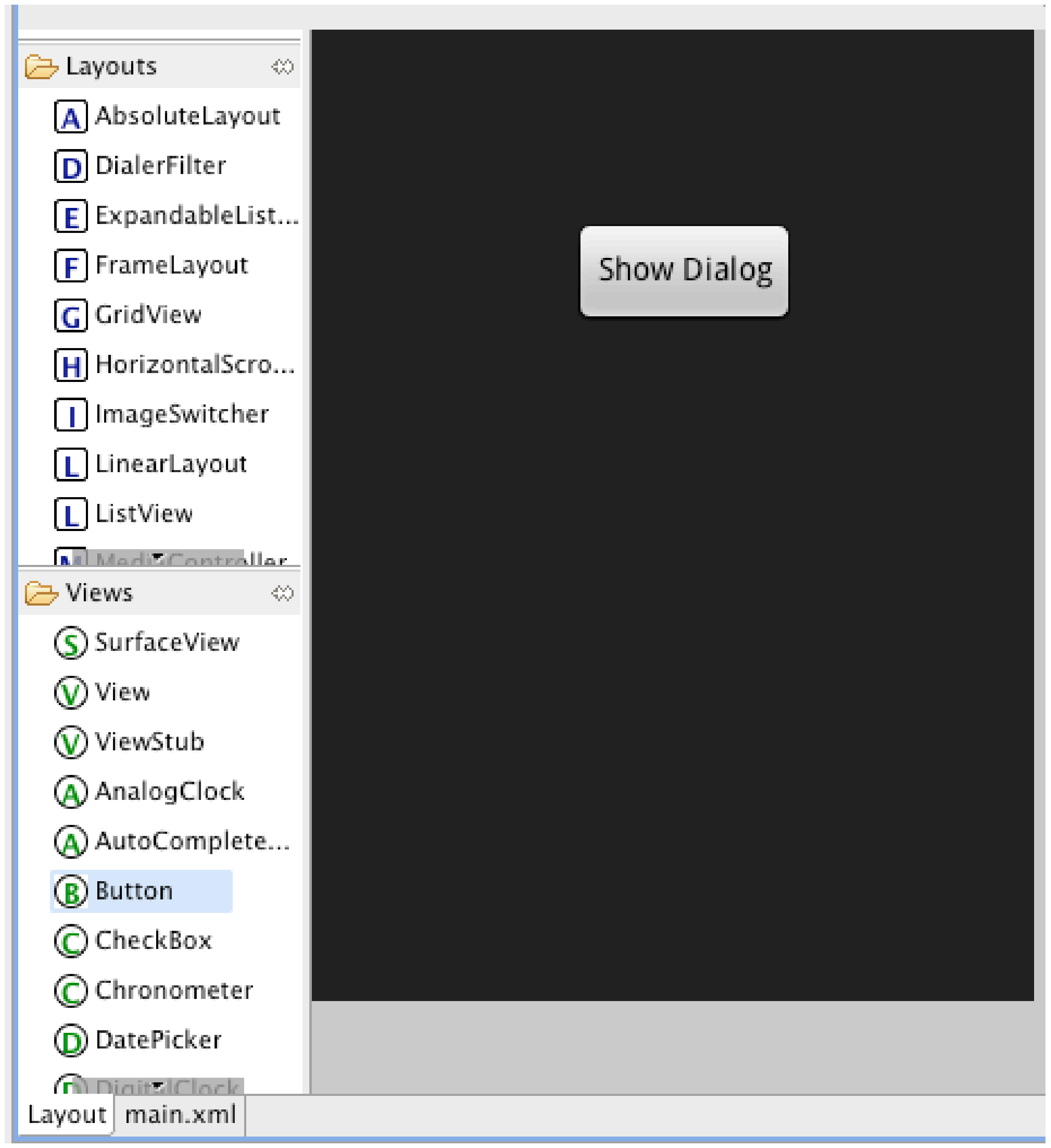
View Attributes

View Attributes

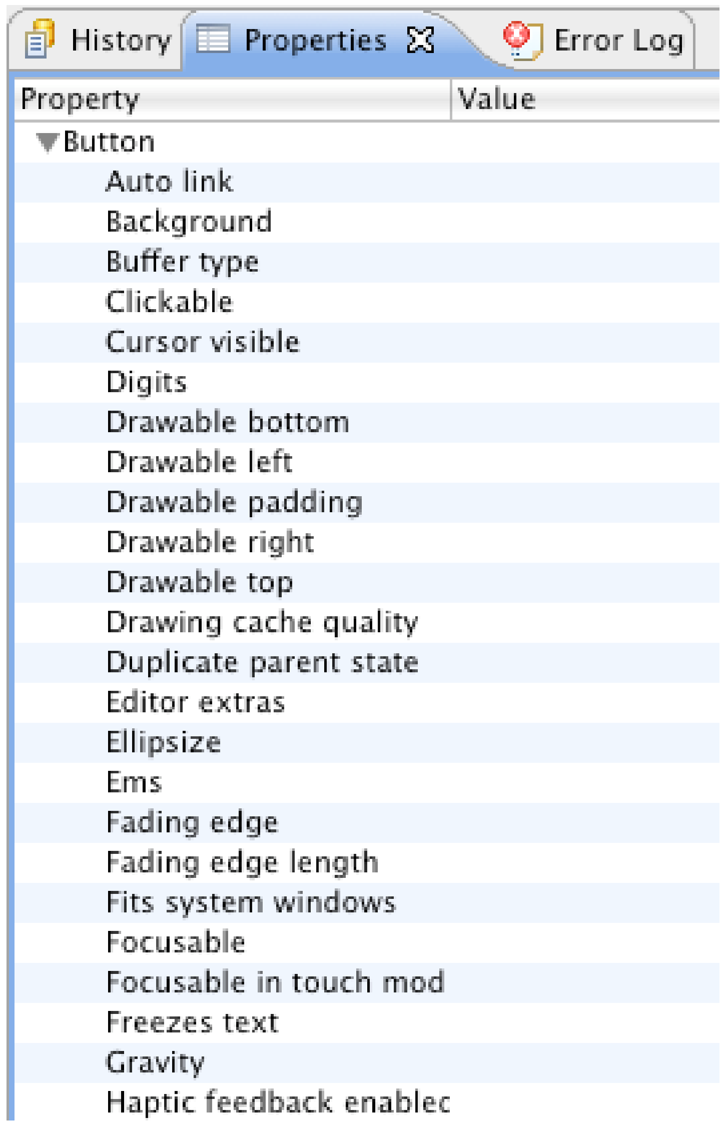
```
<?xml version="1.0" encoding="utf-8"?>

<AbsoluteLayout
    android:id="@+id/AbsoluteLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    xmlns:android="http://schemas.android.com/apk/res/android"
>
    <Button android:text= "Show Dialog"
        android:layout_height="wrap_content"
        android:id="@+id/show"
        android:layout_x="115dip"
        android:layout_width="wrap_content"
        android:layout_y="85dip"
    >
    </Button>
</AbsoluteLayout>
```

Eclipse View Builder



Property Editor



Property	Value
▼Button	
Auto link	
Background	
Buffer type	
Clickable	
Cursor visible	
Digits	
Drawable bottom	
Drawable left	
Drawable padding	
Drawable right	
Drawable top	
Drawing cache quality	
Duplicate parent state	
Editor extras	
Ellipsize	
Ems	
Fading edge	
Fading edge length	
Fits system windows	
Focusable	
Focusable in touch mod	
Freezes text	
Gravity	
Haptic feedback enablec	

View Attributes (Properties)

android:background
android:clickable
android:contentDescription
android:drawingCacheQuality
android:duplicateParentState
android:fadingEdge
android:fadingEdgeLength
android:fitsSystemWindows
android:focusable
android:focusableInTouchMode
android:hapticFeedbackEnabled
android:id
android:isScrollContainer
android:keepScreenOn
android:longClickable
android:minHeight
android:minWidth
android:nextFocusDown
android:nextFocusLeft
android:nextFocusRight
android:nextFocusUp
android:onClick
android:padding
android:paddingBottom
android:paddingLeft
android:paddingRight
android:paddingTop
android:saveEnabled
android:scrollX
android:scrollY
android:scrollbarAlwaysDrawHorizontalTrack
android:scrollbarAlwaysDrawVerticalTrack
android:scrollbarSize
android:scrollbarStyle
android:scrollbarThumbHorizontal
android:scrollbarThumbVertical
android:scrollbarTrackHorizontal
android:scrollbarTrackVertical
android:scrollbars
android:soundEffectsEnabled
android:tag
android:visibility

See http://developer.android.com/reference/android/R.styleable.html#View

TextView Properties

android:autoLink
android:autoText
android:bufferType
android:capitalize
android:cursorVisible
android:digits
android:drawableBottom
android:drawableLeft
android:drawablePadding
android:drawableRight
android:drawableTop
android:editable
android:editorExtras
android:ellipsize
android:ems
android:enabled
android:freezesText
android:gravity
android:height
android:hint
android:imeActionId
android:imeActionLabel
android:imeOptions
android:includeFontPadding
android:inputMethod
android:inputType
android:lineSpacingExtra
android:lineSpacingMultiplier
android:lines

android:linksClickable
android:marqueeRepeatLimit
android:maxEms
android:maxHeight
android:maxLength
android:maxLines
android:maxWidth
android:minEms
android:minHeight
android:minLines
android:minWidth
android:numeric
android:password
android:phoneNumber
android:privateImeOptions
android:scrollHorizontally

android:selectAllOnFocus
android:shadowColor
android:shadowDx
android:shadowDy
android:shadowRadius
android:singleLine
android:text
android:textAppearance
android:textColor
android:textColorHighlight
android:textColorHint
android:textColorLink
android:textScaleX
android:textSize
android:textStyle
android:typeface
android:width

http://developer.android.com/reference/android/R.styleable.html#TextView

TextView_numeric

If set, specifies that this TextView has a numeric input method.
The default is false.
Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
integer	0x01	Input is numeric.
signed	0x003	Input is numeric, with sign allowed.
decimal	0x05	Input is numeric, with decimals allowed.

TextView_digits

TextView has a numeric input method
(This does not seem correct)

Specific characters are the ones that it will accept

Containers - LinearLayout

Important Properties/Concepts

Orientation
Fill Model
Weight
Gravity
Padding

Orientation

android:orientation

horizontal
- view is a row

vertical
- view is a column

Change at runtime

```
setOrientation(LinearLayout.VERTICAL);

setOrientation(LinearLayout.HORIZONTAL);
```

Fill Model

subviews supply
android:layout_width
android:layout_height

Specify
Exact size
wrap_content
fill_parent

Weight

android:layout_weight

Relative weight of views to use in fill_parent

A view of twice the weight take twice the space

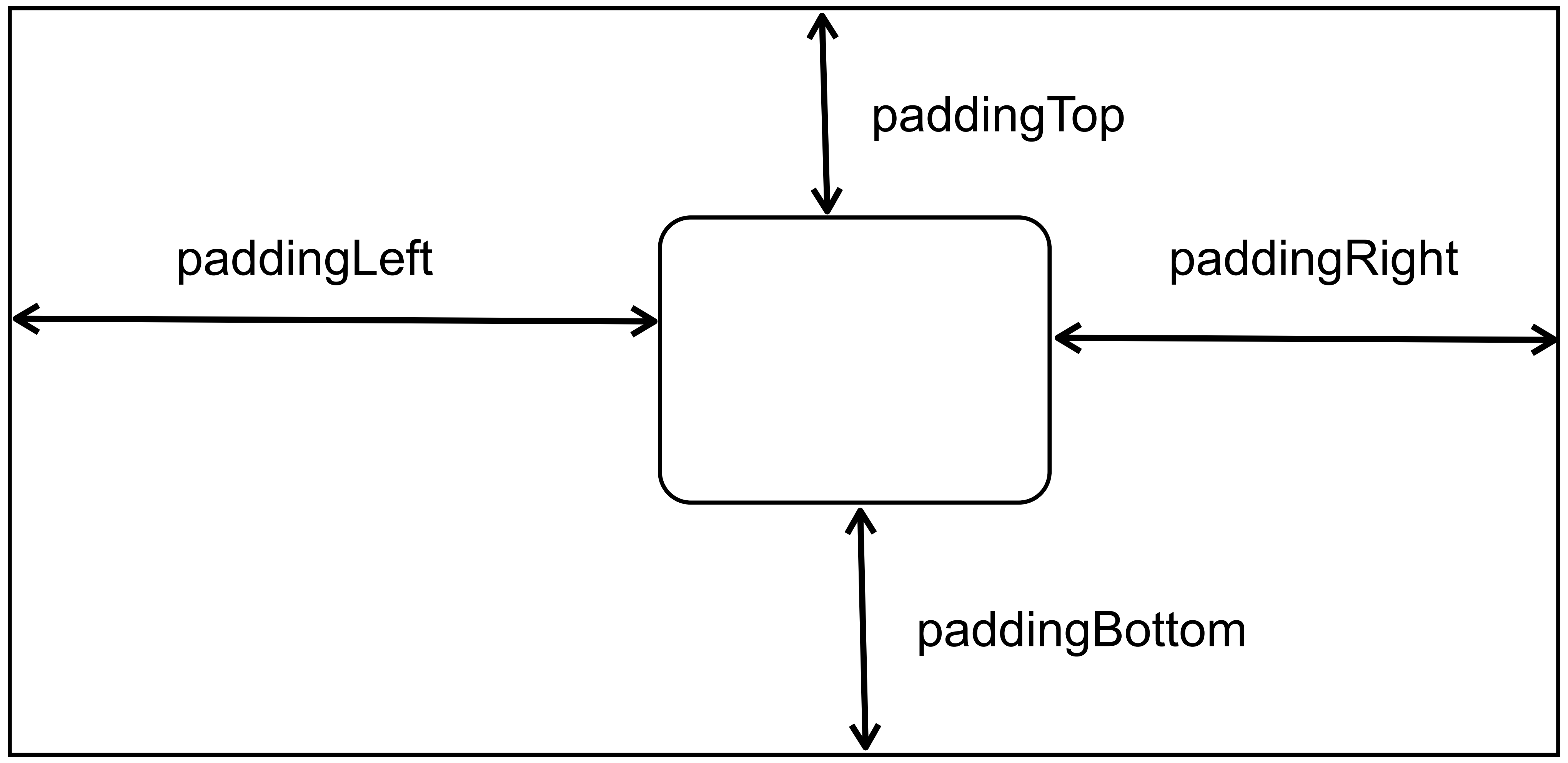
Gravity

android:layout_gravity
setGravity()

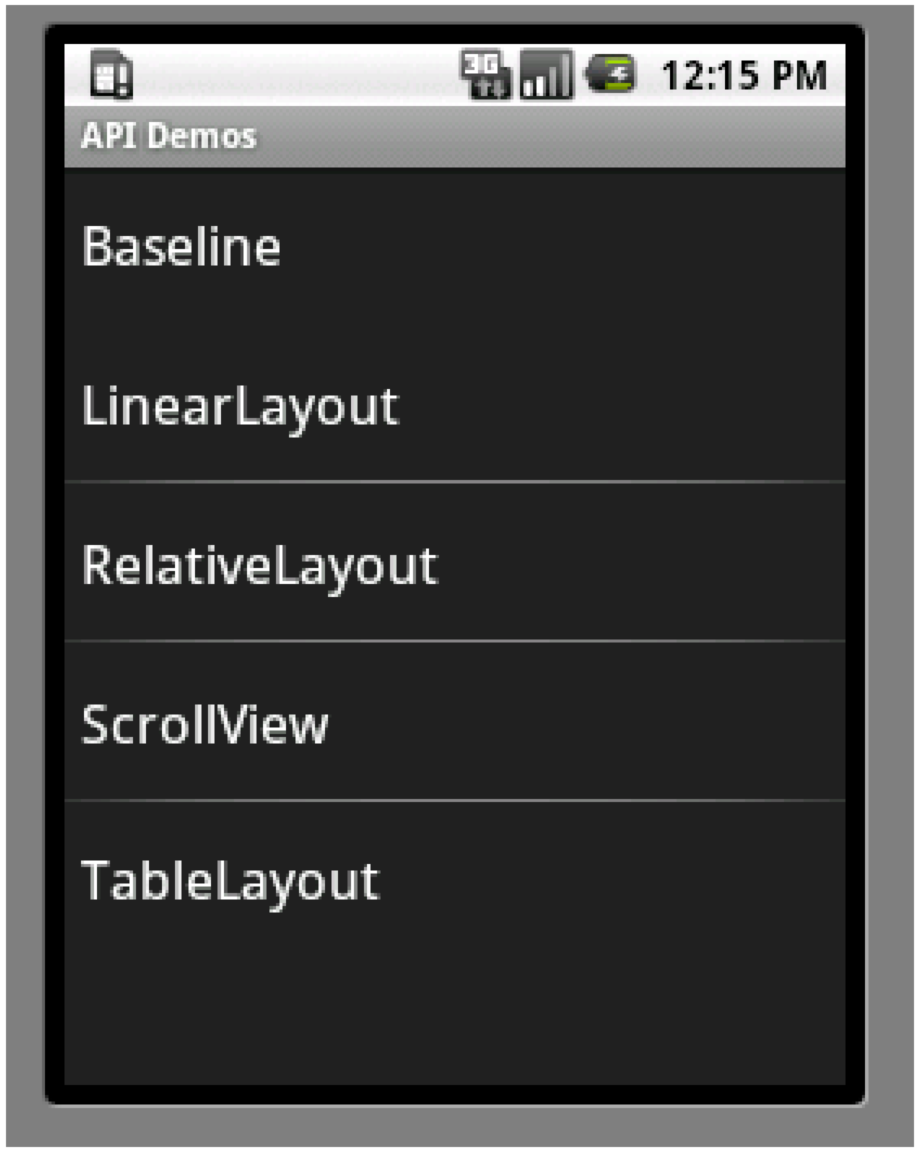
How do the subviews line up

Padding

android:padding
setPadding()



Demo


12:15 PM
API Demos
Baseline
LinearLayout
RelativeLayout
ScrollView
TableLayout

Dialogs

Types of Dialogs

AlertDialog
  - Can have buttons and checkboxes

ProgressDialog

DatePickerDialog

TimePickerDialog

Custom Dialogs

See http://developer.android.com/guide/topics/ui/dialogs.html

Activity.onCreateDialog(int)

Create dialogs in onCreateDialog

```
static final int DIALOG_PAUSED_ID = 0;
static final int DIALOG_GAMEOVER_ID = 1;


protected Dialog onCreateDialog(int id) {
    Dialog dialog;
    switch(id) {
    case DIALOG_PAUSED_ID:
        // do the work to define the pause Dialog
        break;
    case DIALOG_GAMEOVER_ID:
        // do the work to define the game over Dialog
        break;
    default:
        dialog = null;
    }
    return dialog;
}
```

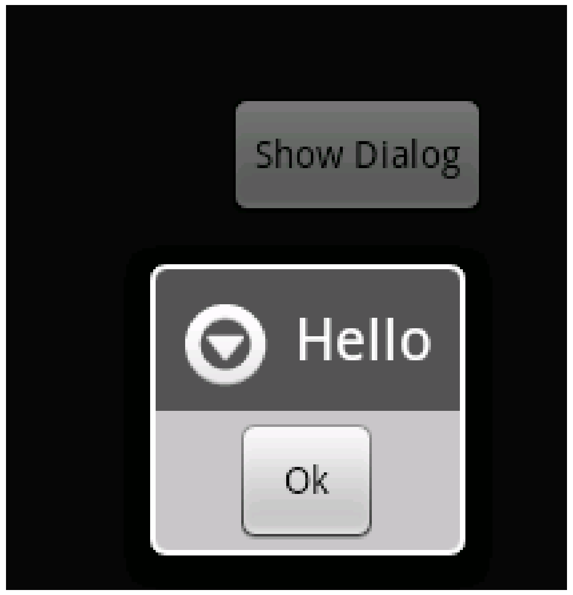
showDialog(int)

To show a dialog in your activity call showDialog(int) which calls onCreateDialog the first time

```
showDialog(DIALOG_PAUSED_ID);
```

Creating an AlertDialog

Class DialogExample



```
protected Dialog onCreateDialog(int id) {
    switch (id) {
    case SAMPLE_DIALOG_ID:
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Hello").setPositiveButton("Ok",
                new DialogInterface.OnClickListener() {
                    public void onClick(DialogInterface dialog,
                            int whichButton) {
                        DialogExample.this.finish();
                        Toast.makeText(getApplicationContext(), "Good Bye",
Toast.LENGTH_SHORT).show();
                    }
                });
        return builder.create();
    default:
        return null;
    }
}
```

Three Buttons

Positive

Negative

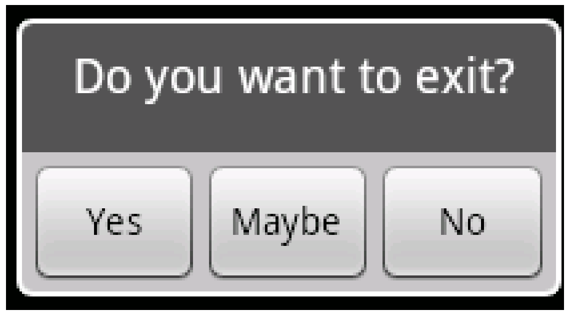
Neutral

Can have only one of each

Button types have no meaing

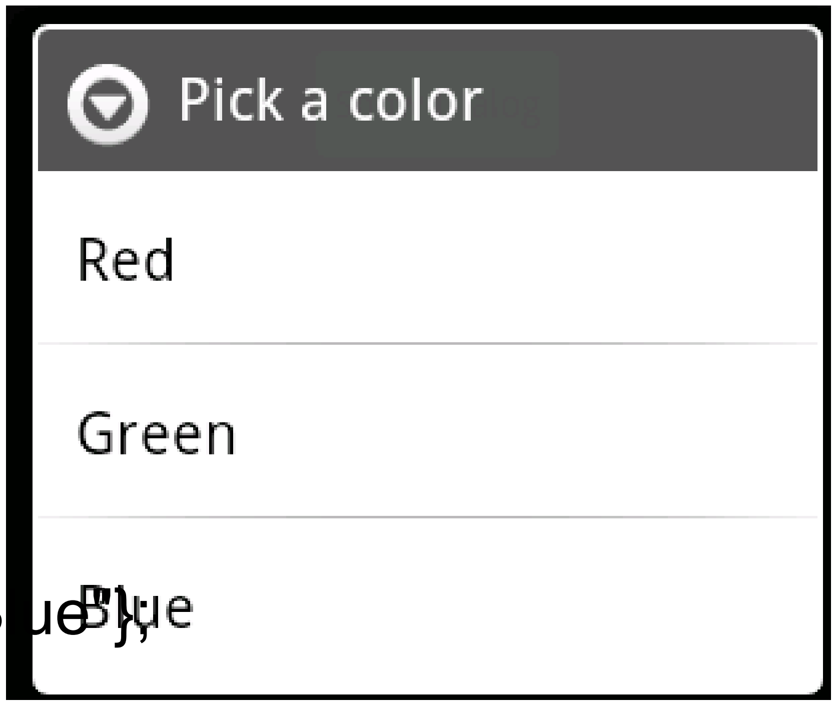
Positive can do what every you want

Three Button Example


Do you want to exit?

```
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setMessage("Do you want to exit?")
    .setCancelable(false)
    .setPositiveButton("Yes",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
                    int whichButton) {
                Toast.makeText(getApplicationContext(), "Good Bye",
                        Toast.LENGTH_SHORT).show();
                DialogExample.this.finish();
            }
        })
        .setNegativeButton("No",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
                    int whichButton) {
                dialog.cancel();
            }
        })
        .setNeutralButton("Maybe",
        new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog,
                    int whichButton) {
                Toast.makeText(getApplicationContext(), "Make up your mind",
                        Toast.LENGTH_SHORT).show();
                DialogExample.this.showDialog(SAMPLE_DIALOG_ID); //Does not work
            }
        });
return builder.create();
```

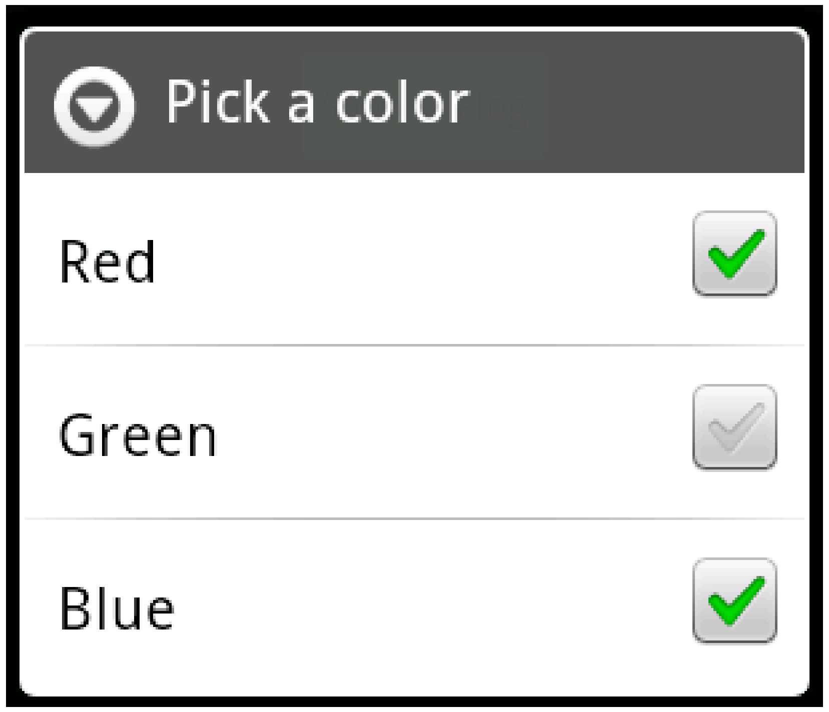
Lists



```
protected Dialog onCreateDialog(int id) {
    switch (id) {
    case SAMPLE_DIALOG_ID:
        final CharSequence[] items = {"Red", "Green", "Blue"};

        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Pick a color");
        builder.setItems(items, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int item) {
                Toast.makeText(getApplicationContext(), items[item],
              Toast.LENGTH_SHORT).show();
            }
        });
        return builder.create();
    default:
        return null;
    }
}
```

MultiSelection



```
protected Dialog onCreateDialog(int id) {
        switch (id) {
        case SAMPLE_DIALOG_ID:
                final CharSequence[] items = {"Red", "Green", "Blue"};
                final boolean[] selected = {false, true, false};

                AlertDialog.Builder builder = new AlertDialog.Builder(this);
                builder.setTitle("Pick a color")
                        .setMultiChoiceItems(items, selected, new DialogInterface.OnMultiChoiceClickListener() {
                    public void onClick(DialogInterface dialog, int item, boolean isChecked) {
                        Toast.makeText(getApplicationContext(), items[item] + " isChecked " + isChecked,
                 Toast.LENGTH_SHORT).show();
                    }
                });
                return builder.create();
        default:
                return null;
        }
}
```

Should add a button to let the user exit. Also need keep track of which items are selected. Docs say that the Dialog has that information - dialog.getListView().isItemChecked(int position) or dialog.getListView().getCheckedItemPositions()

Events

UI Events

Listeners

View.OnClickListener

public void onClick (View v)

Called when a view has been clicked

User either touches the item or

focuses upon the item presses the suitable "enter" key or presses down on the trackball

View.OnLongClickListener

public boolean onLongClick (View v)

Called when a view has been clicked and held for one second.

return True if the callback consumed the long click, false otherwise

View.OnFocusChangeListener

public void onFocusChange (View v, boolean hasFocus)

Called when the focus state of a view has changed.

More Listeners

View.OnKeyListener

public boolean onKey (View v, int keyCode, KeyEvent event)

Called when a key is dispatched to a view

View.OnTouchListener

public boolean onTouch (View v, MotionEvent event)

Called when a touch event is dispatched to a view

MotionEvent

location
Precision of location
Pressure
Size of area pressed
How long the user has pressed
Historical information