FrameLayout

java.lang.Object --> android.view.View --> android.view.ViewGroup --> android.widget.FrameLayout

- `android:foregroundGravity` - Defines the gravity to apply to the foreground drawable.
- `android:measureAllChildren` - Determines whether to measure all children or just those in the VISIBLE or INVISIBLE state when measuring.

LinearLayout

java.lang.Object --> android.view.View --> android.view.ViewGroup --> android.widget.LinearLayout

- `android:baselineAligned` - When set to false, prevents the layout from aligning its children's baselines.
- `android:baselineAlignedChildIndex` - When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align to (that is, which child TextView).
- `android:divider` - Drawable to use as a vertical divider between buttons.
- `android:gravity` - Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
- `android:measureWithLargestChild` - When set to true, all children with a weight will be considered having the minimum size of the largest child.
- `android:orientation` - Should the layout be a column or a row? Use "horizontal" for a row, "vertical" for a column.
- `android:weightSum` - Defines the maximum weight sum.

Relative Layout

Some of the many layout properties available to views in a RelativeLayout include:

- `android:layout_alignParentTop` - If "true", makes the top edge of this view match the top edge of the parent.
- `android:layout_centerVertical` - If "true", centers this child vertically within its parent.
- `android:layout_below` - Positions the top edge of this view below the view specified with a resource ID.
- `android:layout_toRightOf` - Positions the left edge of this view to the right of the view specified with a resource ID.

ScrollView Layout

- `android:fillViewport` - (true/false) Defines whether the scrollview should stretch its content to fill the viewport.

TextView

java.lang.Object --> android.view.View --> android.widget.TextView

- android:allowUndo Whether undo should be allowed for editable text.
- android:autoLink Controls whether links such as urls and email addresses are automatically found and converted to clickable links.

- android:autoSizeMaxTextSize The maximum text size constraint to be used when auto-sizing text.
- android:autoSizeMinTextSize The minimum text size constraint to be used when auto-sizing text.
- android:autoSizePresetSizes Resource array of dimensions to be used in conjunction with autoSizeTextType set to uniform.
- android:autoSizeStepGranularity Specify the auto-size step size if autoSizeTextType is set to uniform.
- android:autoSizeTextType Specify the type of auto-size.
- android:autoText If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
- android:breakStrategy Break strategy (control over paragraph layout).
- android:bufferType Determines the minimum type that getText() will return.
- android:capitalize If set, specifies that this TextView has a textual input method and should automatically capitalize what the user types.
- android:cursorVisible Makes the cursor visible (the default) or invisible.
- android:digits If set, specifies that this TextView has a numeric input method and that these specific characters are the ones that it will accept.
- android:drawableBottom The drawable to be drawn below the text.
- android:drawableEnd The drawable to be drawn to the end of the text.
- android:drawableLeft The drawable to be drawn to the left of the text.
- android:drawablePadding The padding between the drawables and the text.
- android:drawableRight The drawable to be drawn to the right of the text.
- android:drawableStart The drawable to be drawn to the start of the text.
- android:drawableTint Tint to apply to the compound (left, top, etc.) drawables.
- android:drawableTintMode Blending mode used to apply the compound (left, top, etc.) drawables tint.
- android:drawableTop The drawable to be drawn above the text.
- android:editable If set, specifies that this TextView has an input method.
- android:editorExtras Reference to an XML resource containing additional data to supply to an input method, which is private to the implementation of the input method.
- android:elegantTextHeight Elegant text height, especially for less compacted complex script text.
- android:ellipsize If set, causes words that are longer than the view is wide to be ellipsized instead of broken in the middle.
- android:ems Makes the TextView be exactly this many ems wide.
- android:enabled Specifies whether the widget is enabled.
- android:fallbackLineSpacing Whether to respect the ascent and descent of the fallback fonts that are used in displaying the text.
- android:firstBaselineToTopHeight Distance from the top of the TextView to the first text baseline.
- android:focusedSearchResultHighlightColor Color of focused search result highlight.
- android:focusedSearchResultHighlightColor Color of focused search result highlight.
- android:fontFamily Font family (named by string or as a font resource reference) for the text.
- android:fontFeatureSettings Font feature settings.
- android:fontVariationSettings Font variation settings.
- android:freezesText If set, the text view will include its current complete text inside of its frozen icicle in addition to meta-data such as the current cursor position.
- android:gravity Specifies how to align the text by the view's x- and/or y-axis when the text is smaller than the view.
- android:height Makes the TextView be exactly this tall.
- android:hint Hint text to display when the text is empty.

- android:hyphenationFrequency Frequency of automatic hyphenation.
- android:imeActionId Supply a value for EditorInfo.actionId used when an input method is connected to the text view.
- android:imeActionLabel Supply a value for EditorInfo.actionLabel used when an input method is connected to the text view.
- android:imeOptions Additional features you can enable in an IME associated with an editor to improve the integration with your application.
- android:includeFontPadding Leave enough room for ascenders and descenders instead of using the font ascent and descent strictly.
- android:inputMethod If set, specifies that this TextView should use the specified input method (specified by fully-qualified class name).
- android:inputType The type of data being placed in a text field, used to help an input method decide how to let the user enter text.
- android:justificationMode Mode for justification.
- android:lastBaselineToBottomHeight Distance from the bottom of the TextView to the last text baseline.
- android:letterSpacing Text letter-spacing.
- android:lineBreakStyle Specifies the line-break strategies for text wrapping.
- android:lineBreakWordStyle Specifies the line-break word strategies for text wrapping.
- android:lineHeight Explicit height between lines of text.
- android:lineSpacingExtra Extra spacing between lines of text.
- android:lineSpacingMultiplier Extra spacing between lines of text, as a multiplier.
- android:lines Makes the TextView be exactly this many lines tall.
- android:linksClickable If set to false, keeps the movement method from being set to the link movement method even if autoLink causes links to be found.
- android:marqueeRepeatLimit The number of times to repeat the marquee animation.
- android:maxEms Makes the TextView be at most this many ems wide.
- android:maxHeight Makes the TextView be at most this many pixels tall.
- android:maxLength Set an input filter to constrain the text length to the specified number.
- android:maxLines Makes the TextView be at most this many lines tall.
- android:maxWidth Makes the TextView be at most this many pixels wide.
- android:minEms Makes the TextView be at least this many ems wide.
- android:minHeight Makes the TextView be at least this many pixels tall.
- android:minLines Makes the TextView be at least this many lines tall.
- android:minWidth Makes the TextView be at least this many pixels wide.
- android:numeric If set, specifies that this TextView has a numeric input method.
- android:password Whether the characters of the field are displayed as password dots instead of themselves.
- android:phoneNumber If set, specifies that this TextView has a phone number input method.
- android:privateImeOptions An addition content type description to supply to the input method attached to the text view, which is private to the implementation of the input method.
- android:scrollHorizontally Whether the text is allowed to be wider than the view (and therefore can be scrolled horizontally).
- android:searchResultHighlightColor Color of search results highlight.
- android:searchResultHighlightColor Color of search results highlight.
- android:selectAllOnFocus If the text is selectable, select it all when the view takes focus.

- android:shadowColor Place a blurred shadow of text underneath the text, drawn with the specified color.
- android:shadowDx Horizontal offset of the text shadow.
- android:shadowDy Vertical offset of the text shadow.
- android:shadowRadius Blur radius of the text shadow.
- android:singleLine Constrains the text to a single horizontally scrolling line instead of letting it wrap onto multiple lines, and advances focus instead of inserting a newline when you press the enter key.
- android:text Text to display.
- android:textAllCaps Present the text in ALL CAPS.
- android:textAppearance Base text color, typeface, size, and style.
- android:textColor Text color.
- android:textColorHighlight Color of the text selection highlight.
- android:textColorHint Color of the hint text.
- android:textColorLink Text color for links.
- android:textCursorDrawable Reference to a drawable that will be drawn under the insertion cursor.
- android:textFontWeight Weight for the font used in the TextView.
- android:textIsSelectable Indicates that the content of a non-editable text can be selected.
- android:textScaleX Sets the horizontal scaling factor for the text.
- android:textSelectHandle Reference to a drawable that will be used to display a text selection anchor for positioning the cursor within text.
- android:textSelectHandleLeft Reference to a drawable that will be used to display a text selection anchor on the left side of a selection region.
- android:textSelectHandleRight Reference to a drawable that will be used to display a text selection anchor on the right side of a selection region.
- android:textSize Size of the text.
- android:textStyle Style (normal, bold, italic, bold|italic) for the text.
- android:typeface Typeface (normal, sans, serif, monospace) for the text.
- android:width Makes the TextView be exactly this wide.

```
<TextView
    android:id="@+id/myTextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Chào mừng bạn đến với ứng dụng của chúng tôi!"
    android:textSize="18sp"
    android:textColor="@android:color/black"
    android:background="@android:color/white"
    android:padding="16dp"
    android:layout_margin="8dp"
    android:gravity="center"
    android:fontFamily="sans-serif"
    android:textStyle="bold"
    android:lineSpacingExtra="4dp"
    android:maxLines="3"
    android:ellipsize="end"
    android:visibility="visible"
    android:layout_gravity="center"
    android:shadowColor="@android:color/darker_gray"
    android:shadowRadius="1"
    android:shadowDx="1"
    android:shadowDy="1"
/>

textView.text = "Nội dung văn bản"
Định dạng
val spannableString = SpannableString("Nội dung văn bản")
spannableString.setSpan(StyleSpan(Typeface.BOLD), 0, 5,
Spannable.SPAN_EXCLUSIVE_EXCLUSIVE)
textView.text = spannableString

textView.textSize = 20f // Kích thước văn bản tính bằng dp
textView.setTextColor(Color.RED) // Màu sắc văn bản
textView.typeface = Typeface.create("sans-serif", Typeface.BOLD)
textView.gravity = Gravity.CENTER // Căn giữa
textView.layoutParams = LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.WRAP_CONTENT,
    LinearLayout.LayoutParams.WRAP_CONTENT
)

textView.background = ContextCompat.getDrawable(context, R.drawable.border)
textView.setOnClickListener {
    // Xử lý sự kiện click
}
textView.visibility = View.VISIBLE // Hoặc View.GONE, View.INVISIBLE
```

Button

java.lang.Object --> android.view.View --> android.widget.TextView --> android.widget.Button

```
<Button
    android:id="@+id/my_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Click Me"
    android:textColor="@android:color/white"
    android:background="@color/colorPrimary"
    android:padding="16dp"
    android:layout_margin="8dp"
    android:fontFamily="sans-serif"
    android:textSize="16sp"
    android:drawableLeft="@drawable/ic_launcher_foreground"
    android:drawablePadding="8dp"
    android:visibility="visible"
    android:clickable="true"
    android:focusable="true"
    android:layout_gravity="center"
    android:stateListAnimator="@android:animator/selector"
    android:contentDescription="@string/button_description" />
```

```
button.text = "Nhấn vào đây"
button.setTextColor(Color.WHITE) // Màu sắc văn bản
button.setBackgroundColor(Color.BLUE) // Màu nền
button.typeface = Typeface.create("sans-serif", Typeface.BOLD)
button.textSize = 16f // Kích thước văn bản tính bằng sp
button.setOnClickListener {
    // Xử lý sự kiện click
}
button.visibility = View.VISIBLE // Hoặc View.GONE, View.INVISIBLE
button.layoutParams = LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.WRAP_CONTENT,
    LinearLayout.LayoutParams.WRAP_CONTENT
)
button.background = ContextCompat.getDrawable(context, R.drawable.border)
button.isEnabled = true // Hoặc false để vô hiệu hóa
button.setCompoundDrawablesWithIntrinsicBounds(R.drawable.ic_icon, 0, 0, 0) //
Hình ảnh bên trái
val drawable = GradientDrawable()
drawable.cornerRadius = 16f // Độ bo góc
drawable.setColor(Color.BLUE) // Màu nền
button.background = drawable
```

EditText

java.lang.Object --> android.view.View --> android.widget.TextView --> android.widget.EditText

```
<EditText
    android:id="@+id/my_edit_text"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Enter your text here"
    android:inputType="text"
    android:textColor="@android:color/black"
    android:textColorHint="@android:color/darker_gray"
    android:background="@drawable/edit_text_background"
    android:padding="16dp"
    android:layout_margin="8dp"
    android:fontFamily="sans-serif"
    android:textSize="16sp"
    android:maxLines="1"
    android:singleLine="true"
    android:imeOptions="actionDone"
    android:visibility="visible"
    android:focusable="true"
    android:focusableInTouchMode="true"
    android:drawableLeft="@drawable/ic_launcher_foreground"
    android:drawablePadding="8dp"
    android:contentDescription="@string/edit_text_description" />

editText.setText("Nội dung văn bản")
```

```
val text = editText.text.toString()
editText.setTextColor(Color.BLACK) // Màu sắc văn bản
editText.textSize = 16f // Kích thước văn bản tính bằng sp
editText.typeface = Typeface.create("sans-serif", Typeface.NORMAL)
editText.visibility = View.VISIBLE // Hoặc View.GONE, View.INVISIBLE
editText.layoutParams = LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.MATCH_PARENT,
    LinearLayout.LayoutParams.WRAP_CONTENT
)
editText.background = ContextCompat.getDrawable(context, R.drawable.border)
editText.addTextChangedListener(object : TextWatcher {
    override fun beforeTextChanged(s: CharSequence?, start: Int, count: Int,
after: Int) {}
    override fun onTextChanged(s: CharSequence?, start: Int, before: Int, count:
Int) {
        // Xử lý khi văn bản thay đổi
    }
    override fun afterTextChanged(s: Editable?) {}
})
editText.hint = "Nhập văn bản ở đây"
editText.inputType = InputType.TYPE_CLASS_TEXT or
InputType.TYPE_TEXT_VARIATION_PASSWORD // Nhập mật khẩu
editText.filters = arrayOf<InputFilter>(InputFilter.LengthFilter(20)) // Giới hạn
20 ký tự
editText.setOnFocusChangeListener { v, hasFocus ->
    if (hasFocus) {
        // Xử lý khi EditText được chọn
    } else {
        // Xử lý khi EditText mất focus
    }
}
```

Checkbox

java.lang.Object --> android.view.View --> android.widget.TextView --> android.widget.Button --> android.widget.CompoundButton --> android.widget.CheckBox

```

<CheckBox
    android:id="@+id/my_check_box"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Accept Terms and Conditions"
    android:textColor="@android:color/black"
    android:background="?android:attr/selectableItemBackground"
    android:padding="8dp"
    android:layout_margin="8dp"
    android:checked="false"
    android:button="@drawable/custom_checkbox"
    android:visibility="visible"
    android:focusable="true"
    android:contentDescription="@string/checkbox_description" />
```

```
checkBox.text = "Chọn tôi"
val isChecked = checkBox.isChecked
checkBox.isChecked = true // Hoặc false để bỏ chọn
checkBox.setTextColor(Color.BLACK) // Màu sắc văn bản
checkBox.setOnCheckedChangeListener { buttonView, isChecked ->
    if (isChecked) {
        // Xử lý khi CheckBox được chọn
    } else {
        // Xử lý khi CheckBox không được chọn
    }
}
checkBox.isEnabled = false // Để vô hiệu hóa CheckBox
```

Radio Button

java.lang.Object --> android.view.View --> android.view.ViewGroup --> android.widget.LinearLayout --> android.widget.RadioGroup

java.lang.Object --> android.view.View --> android.widget.TextView --> android.widget.Button --> android.widget.CompoundButton --> android.widget.RadioButton

```
<RadioButton
    android:id="@+id/my_radio_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Option 1"
    android:textColor="@android:color/black"
    android:background="?android:attr/selectableItemBackground"
    android:padding="8dp"
    android:layout_margin="8dp"
    android:checked="false"
    android:button="@drawable/custom_radiobutton"
    android:visibility="visible"
    android:focusable="true"
    android:contentDescription="@string/radiobutton_description" />

<RadioGroup
    android:id="@+id/my_radio_group"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <RadioButton
        android:id="@+id/radio_option1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 1" />

    <RadioButton
        android:id="@+id/radio_option2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 2" />

    <RadioButton
        android:id="@+id/radio_option3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Option 3" />
</RadioGroup>
```

ImageView

java.lang.Object --> android.view.View --> android.widget.ImageView

```
<ImageView
    android:id="@+id/my_image_view"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/my_image"  <!-- Hình ảnh hiển thị -->
    android:contentDescription="@string/image_description"  <!-- Mô tả cho người dùng sử dụng công nghệ hỗ trợ -->
    android:scaleType="centerCrop"  <!-- Cách hình ảnh được hiển thị trong ImageView -->
    android:adjustViewBounds="true"  <!-- Điều chỉnh kích thước của ImageView để giữ tỷ lệ hình ảnh -->
    android:layout_margin="16dp"  <!-- Khoảng cách giữa ImageView và các phần tử khác -->
    android:visibility="visible"  <!-- Trạng thái hiển thị của ImageView -->
    android:background="?android:attr/selectableItemBackground"  <!-- Hình nền với hiệu ứng nhấn -->
    android:padding="8dp" />  <!-- Khoảng cách giữa nội dung và biên của ImageView -->

imageView.setImageResource(R.drawable.your_image)
Glide.with(context)
     .load("https://example.com/image.jpg")
     .into(imageView)
val drawable = imageView.drawable
imageView.visibility = View.VISIBLE // Hoặc View.GONE, View.INVISIBLE
imageView.layoutParams = LinearLayout.LayoutParams(
    LinearLayout.LayoutParams.WRAP_CONTENT,
    LinearLayout.LayoutParams.WRAP_CONTENT
)
imageView.background = ContextCompat.getDrawable(context, R.drawable.border)
imageView.alpha = 0.5f // Giá trị từ 0.0 (hoàn toàn trong suốt) đến 1.0 (hoàn toàn không trong suốt)
imageView.scaleType = ImageView.ScaleType.CENTER_CROP // Hoặc các kiểu khác như FIT_CENTER, FIT_XY, v.v.
imageView.setOnClickListener {
    // Xử lý khi ImageView được nhấn
}
//  Thiết lập ảnh từ file
val file = File("/path/to/your/image.jpg")
val bitmap = BitmapFactory.decodeFile(file.absolutePath)
imageView.setImageBitmap(bitmap)
// Từ URL
val uri: Uri = Uri.parse("content://path/to/your/image")
imageView.setImageURI(uri)

val roundedCorners = RoundedBitmapDrawableFactory.create(resources, bitmap)
roundedCorners.cornerRadius = 16f // Đặt bán kính góc
imageView.setImageDrawable(roundedCorners)

imageView.setColorFilter(Color.argb(100, 255, 0, 0)) // Áp dụng màu đỏ với độ
trong suốt
// độ phân giải
val options = BitmapFactory.Options()
options.inSampleSize = 2 // Giảm kích thước hình ảnh xuống một nửa
val bitmap = BitmapFactory.decodeResource(resources, R.drawable.your_image,
options)
imageView.setImageBitmap(bitmap)
```

ImageButton

java.lang.Object --> android.view.View --> android.widget.ImageView --> android.widget.ImageButton

```

<ImageButton
    android:id="@+id/my_image_button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:src="@drawable/my_image"  <!-- Hình ảnh hiển thị trên nút -->
    android:contentDescription="@string/image_button_description"  <!-- Mô tả cho
người dùng sử dụng công nghệ hỗ trợ -->
    android:background="?android:attr/selectableItemBackground"  <!-- Hình nền với
hiệu ứng nhấn -->
    android:padding="8dp"  <!-- Khoảng cách giữa hình ảnh và biên của nút -->
    android:layout_margin="16dp"  <!-- Khoảng cách giữa nút và các phần tử khác --
>
    android:scaleType="centerCrop"  <!-- Cách hình ảnh được hiển thị trong
ImageButton -->
    android:visibility="visible"  <!-- Trạng thái hiển thị của ImageButton -->
    android:adjustViewBounds="true" />  <!-- Điều chỉnh kích thước của ImageButton
để giữ tỷ lệ hình ảnh -->
```