**Android Studio Palette Quick Reference**

The sections are ordered the same as in the Android Studio palette i.e. Layouts, [Widgets](http://www.iexplo.re/android-studio/android-studio-palette-quick-reference.html#widgets), [Text Fields](http://www.iexplo.re/android-studio/android-studio-palette-quick-reference.html#text fields), [Containers](http://www.iexplo.re/android-studio/android-studio-palette-quick-reference.html#containers), [Date and Time](http://www.iexplo.re/android-studio/android-studio-palette-quick-reference.html#date and time), [Expert](http://www.iexplo.re/android-studio/android-studio-palette-quick-reference.html#expert), and [Custom.](http://www.iexplo.re/android-studio/android-studio-palette-quick-reference.html#custom)

|  |  |
| --- | --- |
| **Layouts** | |
| FrameLayout | FrameLayout is designed to block out an area on the screen to display a single item. |
| LinearLayout (Horizontal) | A Layout that arranges its children in a single row. |
| LinearLayout (Vertical) | A Layout that arranges its children in a single column. |
| TableLayout | A Layout that arranges its children into rows and columns. A TableLayout consists of a number of TableRow objects, each defining a row (actually you can have other children, which will be explained below.) TableLayouts do not display border lines for their rows, columns or cells. Each row has zero or more cells; each cell can hold one View object. The table has as many columns as the row with the most cells. A table can leave cells empty.Cells can span columns, as they can in HTML. |
| TableRow | A Layout that arranges its children horizontally. A TableRow should always be used as a child of a TableLayout. If a TableRow’s parent is not a TableLayout, the TableRow will behave as a horizontal LinearLayout |
| GridLayoutAPI 14: Android 4.0 (IceCreamSandwich) | A layout that places its children in a rectangular grid. |
| RelativeLayout | A Layout where the positions of the children can be described in relation to each other or to the parent |
| **Widgets** | |
| Medium Text | Displays text to the user and optionally allows them to edit it |
| Small Text | Displays text to the user and optionally allows them to edit it. |
| Button | Represents a push-button widget. |
| RadioButton | A radio button is a two-states button that can either be checked or unchecked. |
| CheckBox | A radio button is a two-states button that can either be checked or unchecked. |
| Switch API 14 Android 4.0 (IceCreamSandwich) | A Switch is a two-state toggle switch widget that can select between two objects. |
| ToggleButton | Displays checked/unchecked states as a button with a ‘light’ indicator and by default accompanied with the text ‘ON’ or ‘OFF’ |
| ImageButton | Displays a button with an image (instead of text) that can be pressed or clicked by the user. |
| ImageView | Displays an arbitrary image, such as an icon. |
| ProgressBar (Large) | Visual indicator of progress in some operation. |
| ProgressBar (Normal) | Visual indicator of progress in some operation. |
| ProgressBar (Small) | Visual indicator of progress in some operation. |
| ProgressBar (Horizontal) |  |
| SeekBar | A SeekBar is an extension of a ProgressBar that adds a draggable thumb. |
| RatingBar | RatingBar is an extension of SeekBar and ProgressBar that shows a rating in stars. |
| Spinner | A view that displays one child at a time and lets the user pick among them. |
| WebView | A View that displays web pages. This class is the basis upon which you can roll your own web browser or simply display some online content with your Activity. |

|  |  |
| --- | --- |
| **Text Fields** | |
| Plain Text | EditText is a thin veneer over TextView that configures itself to be editable. |
| Person Name | EditText is a thin veneer over TextView that configures itself to be editable. |
| Password | EditText is a thin veneer over TextView that configures itself to be editable. |
| Password (Numeric) | EditText is a thin veneer over TextView that configures itself to be editable. |
| E-mail | EditText is a thin veneer over TextView that configures itself to be editable. |
| Phone | EditText is a thin veneer over TextView that configures itself to be editable. |
| Postal Address | EditText is a thin veneer over TextView that configures itself to be editable. |
| Multiline Text | EditText is a thin veneer over TextView that configures itself to be editable. |
| Time | EditText is a thin veneer over TextView that configures itself to be editable. |
| Date | EditText is a thin veneer over TextView that configures itself to be editable. |
| Number | EditText is a thin veneer over TextView that configures itself to be editable. |
| Number (Signed) | EditText is a thin veneer over TextView that configures itself to be editable. |
| Number (Decimal) | EditText is a thin veneer over TextView that configures itself to be editable. |
| **Containers** | |
| RadioGroup | This class is used to create a multiple-exclusion for a set of radio buttons. Checking one radio button that belongs to a radio group unchecks any previously checked radio button within the same group. |
| ListView | A view that shows items in a vertically scrolling list. The items come from the ListAdapter associated with this view. |
| GridView | A view that shows items in a two-dimensional scrolling grid. The items in the grid come from the ListAdapter associated with this view. |
| ExpandableListView | A view that shows items in a two-dimensional scrolling grid. The items in the grid come from the ListAdapter associated with this view. |
| ScrollView | Layout container for a view hierarchy that can be scrolled by the user, allowing it to be larger than the physical display. ScrollView only supports vertical scrolling. |
| HorizontalScrollView API 3: Android 1.5 (Cupcake) | Layout container for a view hierarchy that can be scrolled by the user, allowing it to be larger than the physical display. |
| SearchView API 11: Android 3.0 (Honeycomb) | A widget that provides a user interface for the user to enter a query and submit a request to a search provider. Shows a list of query suggestions or results, if available, and allows the user to pick a suggestion or result to launch into. |
| TabHost | Container for a tabbed window view. This object holds two children: a set of tab labels that the user clicks to select a specific tab and a FrameLayout object that displays the contents of that page. The individual elements are typically controlled using this container object, rather than setting values on the child elements themselves. |
| ~~SlidingDrawer~~ | **This item is deprecated in version “API 17: Android 4,2 (Jelly Bean)”. This class is not supported anymore. It is recommended you base your own implementation on on the source code for the Android Open Source Project if you must use it in your application.** Sliding drawer hides content out of the screen and allows the user to drag a handle to bring the content on screen. Sliding drawer can be used vertically or horizontally. A special widget composed of two children views: the handle, that the user drags, and the content, attached to the handle and dragged with it. |
| ~~Gallery~~ | **This item is deprecated in version “Android 16: Android 4.1 (Jelly Bean)”. This widget is no longer supported. Other horizontally scrolling components include HorizontalScrollView and ViewPager from the support library.** A view that shows items in a center-locked, horizontally scrolling list. |
| VideoView | Displays a video file. The VideoView class can load images from various sources (such as resources or content providers), takes care of computing its measurement from the video so that it can be used in any layout manager, and provides various display options such as scaling and tinting. |
| ~~TwoLineListItem~~ | **This item is deprecated in version “Android 17: Android 4.2 (Jelly Bean)”. This class can be implemented easily by apps using a RelativeLayout or a LinearLayout.** A view group with two children, intended for use in ListViews. |
| DialerFilter |  |
|  |  |

|  |  |
| --- | --- |
| **Date and Time** | |
| TextClock API 17 “Android 4.2 (Jelly Bean) | This widget displays the current date and/or time as a formatted string. |
| AnalogClock | This widget displays an analog clock with two hands for hours and minutes. |
| ~~DigitalClock~~ | **This item is deprecated in version “Android 17: Android 4.2 (Jelly bean)”. Use a TextClock instead.** Like AnalogClock but digital. Shows seconds. |
| Chronometer | Class that implements a simple timer. |
| DatePicker | This class is a widget for selecting a date. |
| TimePicker | A view for selecting the time of day, either in 24 hour or AM/PM mode. |
| CalendarView API 11: Android 3.0 (Honeycomb) | This class is a calendar widget for displaying and selecting dates. |
| **Expert** | |
| Space API 14: Android 4.0 (IceCreamSandwich) | Space is a lightweight View subclass that may be used to create gaps between components in general purpose layouts. |
| CheckedTextView | An extension to TextView that supports the Checkable interface. |
| QuickContactBadge API 5: Android 4.2 (Eclair) | Widget used to show an image with the standard QuickContact badge and on-click behaviour. |
| ExtractEditText | Specialization of EditText for showing and interacting with the extracted text in a full-screen input method. |
| AutoCompleteTextView | An editable text view that shows completion suggestions automatically while the user is typing. |
| MultiAutoCompleteTextView | An editable text view, extending AutoCompleteTextView, that can show completion suggestions for the substring of the text where the user is typing instead of necessarily for the whole thing. |
| NumberPicker API 11: Android 3.0 (Honeycomb) | A widget that enables a user to select a number from a predefined range. The widget presents an input field and up and down buttons for selecting the current value. Pressing/long-pressing the up and down buttons increments and decrements the current value respectively. |
| ZoomButton |  |
| ZoomControls | The ZoomControls class displays a simple set of controls used for zooming and provides callbacks to register for events. |
| MediaController | A view containing controls for a MediaPlayer. Typically contains the buttons like ‘Play/Pause’, ‘Rewind’, ‘Fast Forward’ and a progress slider. It takes care of synchronizing the controls with the state of the MediaPlayer. |
| GestureOverlayView | A transparent overlay for gesture input that can be placed on top of other widgets or contain other widgets. |
| SurfaceView | Provides a dedicated drawing surface embedded inside of a view hierarchy. You can control the format of this surface and, if you like, its size; the SurfaceView takes care of placing the surface at the correct location on the screen. |
| TextureView | A TextureView can be used to display a content stream. Such a stream can for instance be a video or an OpenGL scene. |
| StackView API 11: Android 3.0 (Honeycomb) | A view that displays its children in a stack and allows users to discretely swipe through the children. |
| ViewStub | A view stub is an invisible, zero-sized View that can be used to lazily inflate layout resources at runtime. |
| ViewAnimator | Container that will perform animations when switching between its views. |
| ViewFlipper | Simple ViewAnimator that will animate between two or more views that have been added to it. Only one child is shown at a time. If requested, can automatically flip between each child at a regular interval. |
| ViewSwitcher | ViewAnimator that switches between two views, and has a factory from which these views are created. You can either use the factory to create the views, or add them yourself. A ViewSwitcher can only have two child views, of which only one is shown at a time. |
| ImageSwitcher |  |
| TextSwitcher | Specialized ViewSwitcher that contains only children of a type TextView. A TextSwitcher is used to animate a label on screen. |
| AdapterViewFlipper API 11: Android 3.0 (Honeycomb) | Simple ViewAnimator that will animate between two or more views that have been added to it. Only one child is shown at a time. If requested, can automatically flip between each child at a regular interval. |

|  |  |
| --- | --- |
| **Custom** | |
| <include> | Lets you statically include XML layout inside other XML layouts. |
| <fragment> | A fragment is a piece of an application’s user interface or behaviour that can be placed in an Activity. |
| RequestFocus | Requests focus from the parent element or one of its descendants. |
| CustomView |  |