**Android Toggle Button** can be used to display checked/unchecked (On/Off) state on the button.

It is beneficial if user have to change the setting between two states. It can be used to On/Off Sound, Wifi, Bluetooth etc.

ToggleButton class provides the facility of creating the toggle button.

XML Attributes of ToggleButton class

The 3 XML attributes of ToggleButton class.

|  |  |
| --- | --- |
| **XML Attribute** | **Description** |
| android:disabledAlpha | The alpha to apply to the indicator when disabled. |
| android:textOff | The text for the button when it is not checked. |
| android:textOn | The text for the button when it is checked. |

Methods of ToggleButton class

The widely used methods of ToggleButton class are given below.

|  |  |
| --- | --- |
| **Method** | **Description** |
| CharSequence getTextOff() | Returns the text when button is not in the checked state. |
| CharSequence getTextOn() | Returns the text for when button is in the checked state. |
| void setChecked(boolean checked) | Changes the checked state of this button. |

You can add a basic toggle button to your layout with the [ToggleButton](https://developer.android.com/reference/android/widget/ToggleButton) object. Android 4.0 (API level 14) introduces another kind of toggle button called a switch that provides a slider control, which you can add with a [Switch](https://developer.android.com/reference/android/widget/Switch) object. [SwitchCompat](https://developer.android.com/reference/androidx/appcompat/widget/SwitchCompat) is a version of the Switch widget which runs on devices back to API 7.

If you need to change a button's state yourself, you can use the [CompoundButton.setChecked()](https://developer.android.com/reference/android/widget/CompoundButton#setChecked(boolean)) or [CompoundButton.toggle()](https://developer.android.com/reference/android/widget/CompoundButton#toggle()) method.

https://developer.android.com/images/ui/togglebutton.png

Toggle buttons

https://developer.android.com/images/ui/switch.png

Switches (in Android 4.0+)

Key classes are the following:

* [ToggleButton](https://developer.android.com/reference/android/widget/ToggleButton)
* [Switch](https://developer.android.com/reference/android/widget/Switch)
* [SwitchCompat](https://developer.android.com/reference/androidx/appcompat/widget/SwitchCompat)
* [CompoundButton](https://developer.android.com/reference/android/widget/CompoundButton)

## Responding to Button Presses

To detect when the user activates the button or switch, create an [CompoundButton.OnCheckedChangeListener](https://developer.android.com/reference/android/widget/CompoundButton.OnCheckedChangeListener) object and assign it to the button by calling [setOnCheckedChangeListener()](https://developer.android.com/reference/android/widget/CompoundButton" \l "setOnCheckedChangeListener(android.widget.CompoundButton.OnCheckedChangeListener)). For example:

ToggleButton toggle = (ToggleButton) findViewById(R.id.togglebutton);  
toggle.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
    public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
        if (isChecked) {  
            // The toggle is enabled  
        } else {  
            // The toggle is disabled  
        }  
    }  
}

**Switch**

A Switch is a two-state toggle switch widget that can select between two options. The user may drag the "thumb" back and forth to choose the selected option, or simply tap to toggle as if it were a checkbox. The [text](https://developer.android.com/reference/android/widget/TextView#setText(java.lang.CharSequence)) property controls the text displayed in the label for the switch, whereas the [off](https://developer.android.com/reference/android/widget/Switch#setTextOff(java.lang.CharSequence)) and [on](https://developer.android.com/reference/android/widget/Switch#setTextOn(java.lang.CharSequence)) text controls the text on the thumb.

|  |  |
| --- | --- |
| [aandroid:textOff](https://developer.android.com/reference/android/widget/Switch#attr_android:textOff) | Text to use when the switch is in the unchecked/"off" state. |
| [aandroid:textOn](https://developer.android.com/reference/android/widget/Switch#attr_android:textOn) | Text to use when the switch is in the checked/"on" state. |

**ChipGroup**

Chips are compact elements that represent an attribute, text, entity, or action. They allow users to enter information, select a choice, filter content, or trigger an action.

A ChipGroup is used to hold multiple Chips. By default, the chips are reflowed across multiple lines. Set the app:singleLine attribute to constrain the chips to a single horizontal line. If you do so, you'll usually want to wrap this ChipGroup in a HorizontalScrollView.

ChipGroup also supports a multiple-exclusion scope for a set of chips. When you set the app:singleSelection attribute, checking one chip that belongs to a chip group unchecks any previously checked chip within the same group.

Chips are a newer and stylised form of RadioButtons.

Chips are compact elements that represent an attribute, text, entity, or action. They allow users to enter information, select a choice, filter content, or trigger an action.

The Chip widget is a thin view wrapper around the [ChipDrawable](https://developer.android.com/reference/com/google/android/material/chip/ChipDrawable), which contains all of the layout and draw logic. The extra logic exists to support touch, mouse, keyboard, and accessibility navigation. The main chip and close icon are considered to be separate logical sub-views, and contain their own navigation behavior and state.

All attributes from [R.styleable.Chip](https://developer.android.com/reference/com/google/android/material/R.styleable" \l "Chip) are supported. Do not use the android:background attribute. It will be ignored because Chip manages its own background Drawable. Also do not use the android:drawableStart and android:drawableEnd attributes. They will be ignored because Chip manages its own start (app:chipIcon) and end (app:closeIcon) drawables. The basic attributes you can set are:

* [android:checkable](https://developer.android.com/reference/android/R.attr#checkable) - If true, the chip can be toggled. If false, the chip acts like a button.
* [android:text](https://developer.android.com/reference/android/R.attr#text) - Sets the text of the chip.
* [app:chipIcon](https://developer.android.com/reference/com/google/android/material/R.attr#chipIcon) and [app:chipIconEnabled](https://developer.android.com/reference/com/google/android/material/R.attr" \l "chipIconEnabled) - Sets the icon of the chip. Usually on the left.
* [app:checkedIcon](https://developer.android.com/reference/com/google/android/material/R.attr#checkedIcon) and [app:checkedIconEnabled](https://developer.android.com/reference/com/google/android/material/R.attr" \l "checkedIconEnabled) - Sets a custom icon to use when checked. Usually on the left.
* [app:closeIcon](https://developer.android.com/reference/com/google/android/material/R.attr#closeIcon) and [app:closeIconEnabled](https://developer.android.com/reference/com/google/android/material/R.attr" \l "closeIconEnabled) - Sets a custom icon that the user can click to close. Usually on the right.

You can register a listener on the main chip with [View.setOnClickListener(OnClickListener)](https://developer.android.com/reference/android/view/View#setOnClickListener(android.view.View.OnClickListener)) or [CompoundButton.setOnCheckedChangeListener(OnCheckedChangeListener)](https://developer.android.com/reference/android/widget/CompoundButton#setOnCheckedChangeListener(android.widget.CompoundButton.OnCheckedChangeListener)). You can register a listener on the close icon with [setOnCloseIconClickListener(OnClickListener)](https://developer.android.com/reference/com/google/android/material/chip/Chip" \l "setOnCloseIconClickListener(android.view.View.OnClickListener)).

For proper rendering of the ancestor TextView in RTL mode, call [setLayoutDirection(int)](https://developer.android.com/reference/com/google/android/material/chip/Chip" \l "setLayoutDirection(int)) with View.LAYOUT\_DIRECTION\_LOCALE. By default, TextView's layout rendering sets the text padding in LTR on initial rendering and it only renders correctly after the layout has been invalidated so you need to ensure that initial rendering has the correct layout.

ChipDrawable contains all the layout and draw logic for [Chip](https://developer.android.com/reference/com/google/android/material/chip/Chip).

You can use ChipDrawable directly in contexts that require a Drawable. For example, an auto-complete enabled EditText can replace snippets of text with a ChipDrawable to represent it as a semantic entity. To create an instance of ChipDrawable,

use [createFromResource(Context, int)](https://developer.android.com/reference/com/google/android/material/chip/ChipDrawable" \l "createFromResource(android.content.Context,%20int)) and pass in an XML resource in this form:

<chip xmlns:app="http://schemas.android.com/apk/res-auto"

android:text="Hello, World!"/>

The basic attributes you can set are:

* [android:checkable](https://developer.android.com/reference/android/R.attr#checkable) - If true, the chip can be toggled. If false, the chip acts like a button.
* [android:text](https://developer.android.com/reference/android/R.attr#text) - Sets the text of the chip.
* [app:chipIcon](https://developer.android.com/reference/com/google/android/material/R.attr#chipIcon) - Sets the icon of the chip, or use @null to display no icon. Usually on the left.
* [app:checkedIcon](https://developer.android.com/reference/com/google/android/material/R.attr#checkedIcon) - Sets a custom icon to use when checked, or use @null to display no icon. Usually on the left.
* [app:closeIcon](https://developer.android.com/reference/com/google/android/material/R.attr#closeIcon) - Sets a custom icon that the user can click to close, or use @null to display no icon. Usually on the right.
* [R.attr.ellipsize](https://developer.android.com/reference/android/R.attr#ellipsize) –Doesnot support  [TextUtils.TruncateAt.MARQUEE](https://developer.android.com/reference/android/text/TextUtils.TruncateAt" \l "MARQUEE) because chip text should not scroll.

When used in this stand-alone mode, the host view must explicitly manage the ChipDrawable's state:

* [Drawable.setBounds(int, int, int, int)](https://developer.android.com/reference/android/graphics/drawable/Drawable#setBounds(int,%20int,%20int,%20int)), taking into account [getIntrinsicHeight()](https://developer.android.com/reference/com/google/android/material/chip/ChipDrawable" \l "getIntrinsicHeight()) and [getIntrinsicWidth()](https://developer.android.com/reference/com/google/android/material/chip/ChipDrawable" \l "getIntrinsicWidth()).
* [draw(Canvas)](https://developer.android.com/reference/com/google/android/material/chip/ChipDrawable#draw(android.graphics.Canvas))
* [Drawable.setCallback(Callback)](https://developer.android.com/reference/android/graphics/drawable/Drawable#setCallback(android.graphics.drawable.Drawable.Callback)), to support invalidations on the chip drawable or any of its child drawables. This includes animations.
* [Drawable.setState(int[])](https://developer.android.com/reference/android/graphics/drawable/Drawable#setState(int[])), to support checking the chip, and touch/mouse/keyboard interactions on the chip.
* [setCloseIconState(int[])](https://developer.android.com/reference/com/google/android/material/chip/ChipDrawable#setCloseIconState(int[])), to support touch, mouse, or keyboard interactions on the close icon.
* [Drawable.setHotspot(float, float)](https://developer.android.com/reference/android/graphics/drawable/Drawable#setHotspot(float,%20float))
* [Drawable.setLayoutDirection(int)](https://developer.android.com/reference/android/graphics/drawable/Drawable#setLayoutDirection(int)), to support RTL mode.

ChipDrawable contains three child drawables: chipIcon, checkedIcon, and closeIcon. chipIcon and checkedIcon inherit the state of this drawable, but closeIcon contains its own state that you can set with [setCloseIconState(int[])](https://developer.android.com/reference/com/google/android/material/chip/ChipDrawable" \l "setCloseIconState(int[])).

A ChipGroup is used to hold multiple [Chip](https://developer.android.com/reference/com/google/android/material/chip/Chip)s. By default, the chips are reflowed across multiple lines. Set the [app:singleLine](https://developer.android.com/reference/com/google/android/material/R.attr" \l "singleLine) attribute to constrain the chips to a single horizontal line. If you do so, you'll usually want to wrap this ChipGroup in a [HorizontalScrollView](https://developer.android.com/reference/android/widget/HorizontalScrollView).

ChipGroup also supports a multiple-exclusion scope for a set of chips. When you set the [app:singleSelection](https://developer.android.com/reference/com/google/android/material/R.attr" \l "singleSelection) attribute, checking one chip that belongs to a chip group unchecks any previously checked chip within the same group. The behavior mirrors that of [RadioGroup](https://developer.android.com/reference/android/widget/RadioGroup).

|  |  |
| --- | --- |
| Public methods | |
| void | [addView](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#addView(android.view.View,%20int,%20android.view.ViewGroup.LayoutParams))(View child, int index, ViewGroup.LayoutParams params) |
| void | [check](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#check(int))(int id)  Sets the selection to the chip whose identifier is passed in parameter. |
| void | [clearCheck](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#clearCheck())()  Clears the selection. |
| ViewGroup.LayoutParams | [generateLayoutParams](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#generateLayoutParams(android.util.AttributeSet))(AttributeSet attrs) |
| int | [getCheckedChipId](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#getCheckedChipId())()  When in [single selection mode](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#isSingleSelection()), returns the identifier of the selected chip in this group. |
| List<Integer> | [getCheckedChipIds](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#getCheckedChipIds())()  Returns the identifiers of the selected [Chip](https://developer.android.com/reference/com/google/android/material/chip/Chip)s in this group. |
| int | [getChipSpacingHorizontal](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#getChipSpacingHorizontal())()  Returns the horizontal spacing between visible chips in this group. |
| int | [getChipSpacingVertical](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#getChipSpacingVertical())()  Returns the vertical spacing between visible chips in this group. |
| int | [getRowIndex](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#getRowIndex(android.view.View))(View child)  Gets the row index of the child, primarily for accessibility. |
| boolean | [isSelectionRequired](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#isSelectionRequired())()  Returns whether we prevent all child chips from being deselected. |
| boolean | [isSingleLine](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#isSingleLine())()  Returns whether this chip group is single line or reflowed multiline. |
| boolean | [isSingleSelection](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#isSingleSelection())()  Returns whether this chip group only allows a single chip to be checked. |
| Void | [onInitializeAccessibilityNodeInfo](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#onInitializeAccessibilityNodeInfo(android.view.accessibility.AccessibilityNodeInfo))(AccessibilityNodeInfo info) |
| Void | [setChipSpacing](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setChipSpacing(int))(int chipSpacing)  Sets the horizontal and vertical spacing between visible chips in this group. |
| Void | [setChipSpacingHorizontal](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setChipSpacingHorizontal(int))(int chipSpacingHorizontal)  Sets the horizontal spacing between visible chips in this group. |
| Void | [setChipSpacingHorizontalResource](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setChipSpacingHorizontalResource(int))(int id)  Sets the horizontal spacing between visible chips in this group. |
| Void | [setChipSpacingResource](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setChipSpacingResource(int))(int id)  Sets the horizontal and vertical spacing between visible chips in this group. |
| Void | [setChipSpacingVertical](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setChipSpacingVertical(int))(int chipSpacingVertical)  Sets the vertical spacing between visible chips in this group. |
| Void | [setChipSpacingVerticalResource](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setChipSpacingVerticalResource(int))(int id)  Sets the vertical spacing between visible chips in this group. |
| Void | [setDividerDrawableHorizontal](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setDividerDrawableHorizontal(android.graphics.drawable.Drawable))(Drawable divider)  This method is deprecated. Use *[setChipSpacingHorizontal(int)](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup" \l "setChipSpacingHorizontal(int))* instead. |
| Void | [setDividerDrawableVertical](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setDividerDrawableVertical(android.graphics.drawable.Drawable))(Drawable divider)  This method is deprecated. Use *[setChipSpacingVertical(int)](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup" \l "setChipSpacingVertical(int))* instead. |
| Void | [setFlexWrap](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setFlexWrap(int))(int flexWrap)  This method is deprecated. Use *[setSingleLine(int)](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup" \l "setSingleLine(int))* instead. |
| Void | [setOnCheckedChangeListener](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setOnCheckedChangeListener(com.google.android.material.chip.ChipGroup.OnCheckedChangeListener))([ChipGroup.OnCheckedChangeListener](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup.OnCheckedChangeListener) listener)  Register a callback to be invoked when the checked chip changes in this group. |
| Void | [setOnHierarchyChangeListener](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setOnHierarchyChangeListener(android.view.ViewGroup.OnHierarchyChangeListener))(ViewGroup.OnHierarchyChangeListener listener) |
| Void | [setSelectionRequired](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setSelectionRequired(boolean))(boolean selectionRequired)  Sets whether we prevent all child chips from being deselected. |
| Void | [setShowDividerHorizontal](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setShowDividerHorizontal(int))(int dividerMode)  This method is deprecated. Use *[setChipSpacingHorizontal(int)](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup" \l "setChipSpacingHorizontal(int))* instead. |
| Void | [setShowDividerVertical](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setShowDividerVertical(int))(int dividerMode)  This method is deprecated. Use *[setChipSpacingVertical(int)](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup" \l "setChipSpacingVertical(int))* instead. |
| Void | [setSingleLine](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setSingleLine(boolean))(boolean singleLine)  Sets whether this chip group is single line, or reflowed multiline. |
| void | [setSingleLine](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setSingleLine(int))(int id)  Sets whether this chip group is single line, or reflowed multiline. |
| void | [setSingleSelection](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setSingleSelection(boolean))(boolean singleSelection)  Sets whether this chip group only allows a single chip to be checked. |
| void | [setSingleSelection](https://developer.android.com/reference/com/google/android/material/chip/ChipGroup#setSingleSelection(int))(int id)  Sets whether this chip group only allows a single chip to be checked. |