# VideoView

added in API level 1 (https://developer.android.com/quide/topics/manifest/uses-sdk-element.html#ApiLevels)

public class VideoView

extends <u>SurfaceView</u>(https://developer.android.com/reference/android/view/SurfaceView.html) implements

<u>MediaController.MediaPlayerControl</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html)

java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)

- 4 android.view.View (https://developer.android.com/reference/android/view/View.html)
  - 4 android.view.SurfaceView (https://developer.android.com/reference/android/view/SurfaceView.html)
    - → android.widget.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.

Note: VideoView does not retain its full state when going into the background. In particular, it does not restore the current play state, play position, selected tracks, or any subtitle tracks added via <a href="mailto:addSubtitleSource">addSubtitleSource</a>()

(https://developer.android.com/reference/android/widget/VideoView.html # add Subtitle Source (java.io.Input Stream, % 20 android.media.Media Format)).

Applications should save and restore these on their own in <a href="Activity.onSaveInstanceState(Bundle)">Activity.onSaveInstanceState(Bundle)</a>

(https://developer.android.com/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)) and

# <u>Activity.onRestoreInstanceState(Bundle)</u>

(https://developer.android.com/reference/android/app/Activity.html#onRestoreInstanceState(android.os.Bundle)).

Also note that the audio session id (from getAudioSessionId()

(https://developer.android.com/reference/android/widget/VideoView.html#getAudioSessionId())) may change from its previously returned value when the VideoView is restored.

# By default, VideoView requests audio focus with <a href="mailto:AudioManager.AUDIOFOCUS\_GAIN">AudioManager.AUDIOFOCUS\_GAIN</a>

(https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS\_GAIN). Use <a href="mailto:setAudioFocusRequest(int">setAudioFocusRequest(int)</a>) (https://developer.android.com/reference/android/widget/VideoView.html#setAudioFocusRequest(int)) to change this behavior.

The default <u>AudioAttributes</u> (https://developer.android.com/reference/android/media/AudioAttributes.html) used during playback have a usage of <u>AudioAttributes.USAGE\_MEDIA</u> (https://developer.android.com/reference/android/media/AudioAttributes.html#USAGE\_MEDIA) and a content type of <u>AudioAttributes.CONTENT\_TYPE\_MOVIE</u>

(https://developer.android.com/reference/android/media/AudioAttributes.html#CONTENT\_TYPE\_MOVIE), use <a href="mailto:setAudioAttributes">setAudioAttributes</a>(AudioAttributes)

(https://developer.android.com/reference/android/widget/VideoView.html#setAudioAttributes(android.media.AudioAttributes)) to modify them.

# Summary

# Inherited XML attributes

From class <a href="mailto:android.view.View">android.view.View</a>. (https://developer.android.com/reference/android/view/View.html)

android:accessibilityHeading (https://developer.android.com/reference/android/view/View.html#attr\_android:accessibilityHeading)

android:accessibilityLiveRegion (https://developer.android.com/reference/android/view/View.html#attr\_android:accessibilityLiveRegion)

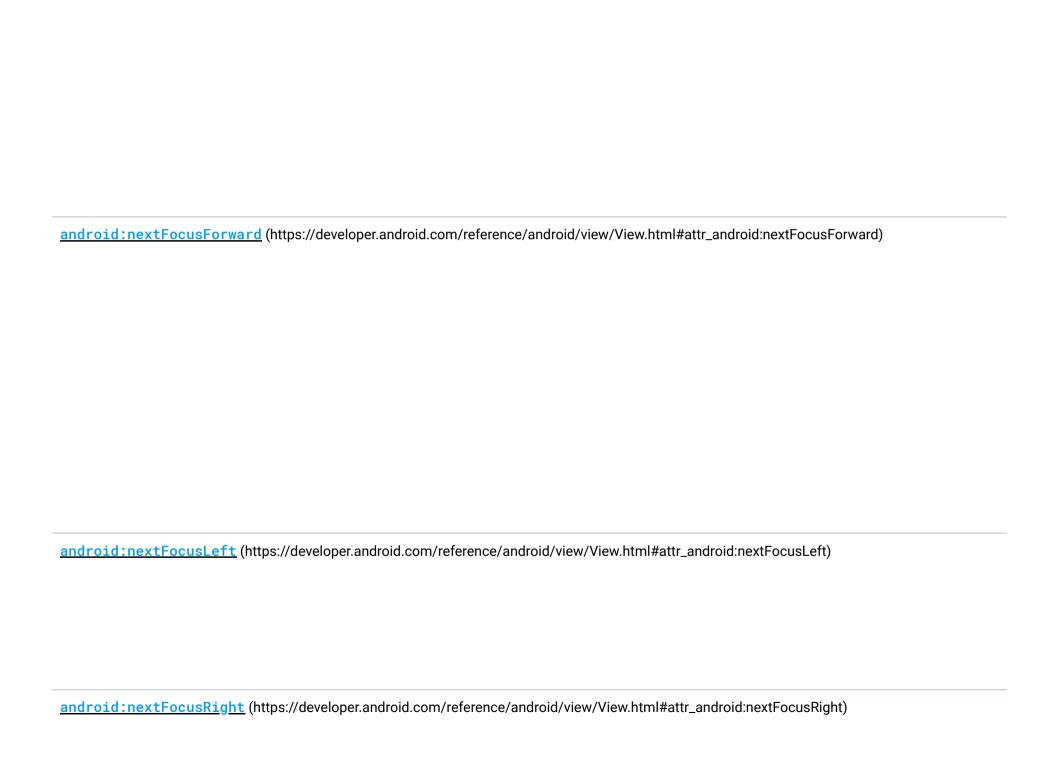
android:accessibilityTraversalAfter (https://developer.android.com/reference/android/view/View.html#attr_android:accessibilityTraversalAfter)
android:accessibilityTraversalBefore (https://developer.android.com/reference/android/view/View.html#attr_android:accessibilityTraversalBefore)
android:alpha (https://developer.android.com/reference/android/view/View.html#attr_android:alpha)
<u>android:autofillHints</u> (https://developer.android.com/reference/android/view/View.html#attr_android:autofillHints)
android:autofilledHighlight (https://developer.android.com/reference/android/view/View.html#attr_android:autofilledHighlight)
android:background (https://developer.android.com/reference/android/view/View.html#attr_android:background)
<pre>android:backgroundTint (https://developer.android.com/reference/android/view/View.html#attr_android:backgroundTint)</pre>
android:backgroundTintMode (https://developer.android.com/reference/android/view/View.html#attr_android:backgroundTintMode)

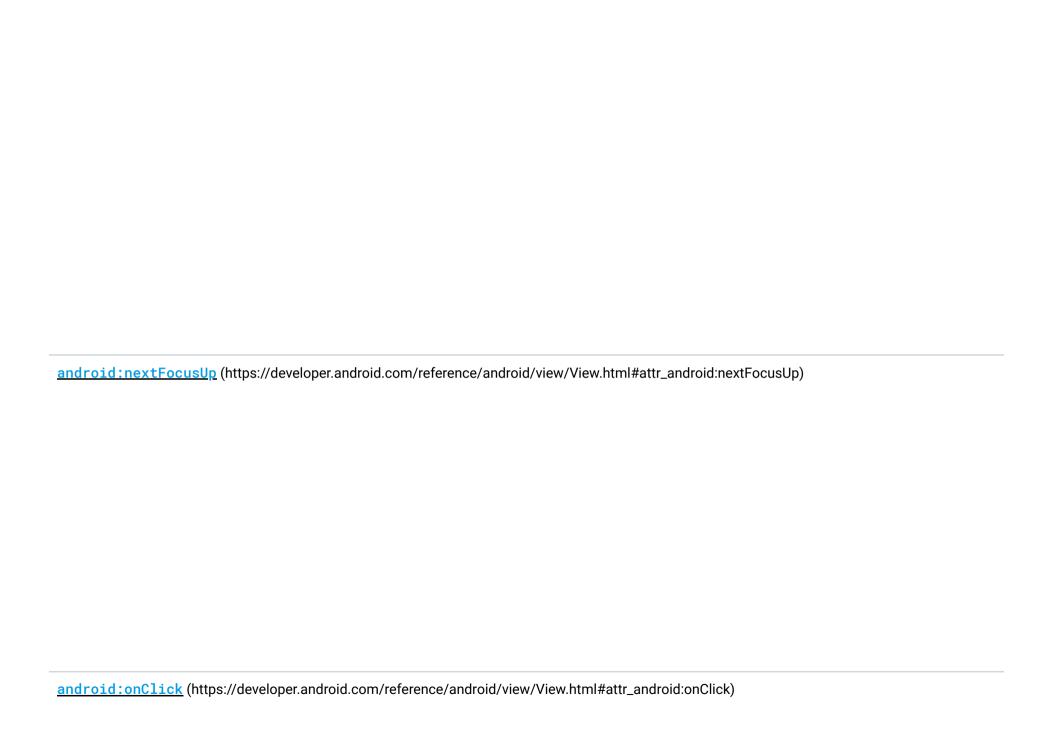
android:clickable (https://developer.android.com/reference/android/view/View.html#attr_android:clickable)
<pre>android:contentDescription (https://developer.android.com/reference/android/view/View.html#attr_android:contentDescription)</pre>
<pre>android:contextClickable (https://developer.android.com/reference/android/view/View.html#attr_android:contextClickable)</pre>
android:defaultFocusHighlightEnabled (https://developer.android.com/reference/android/view/View.html#attr_android:defaultFocusHighlightEnabled)
<u>android:drawingCacheQuality</u> (https://developer.android.com/reference/android/view/View.html#attr_android:drawingCacheQuality)
android:duplicateParentState (https://developer.android.com/reference/android/view/View.html#attr_android:duplicateParentState)
<pre>android:elevation (https://developer.android.com/reference/android/view/View.html#attr_android:elevation)</pre>
<u>android:fadeScrollbars</u> (https://developer.android.com/reference/android/view/View.html#attr_android:fadeScrollbars)
<pre>android:fadingEdgeLength (https://developer.android.com/reference/android/view/View.html#attr_android:fadingEdgeLength)</pre>

android:filterTouchesWhenObscured (https://developer.android.com/reference/android/view/View.html#attr_android:filterTouchesWhenObscure
android:fitsSystemWindows (https://developer.android.com/reference/android/view/View.html#attr_android:fitsSystemWindows)
<pre>android:focusable (https://developer.android.com/reference/android/view/View.html#attr_android:focusable)</pre>
android: focusableInTouchMode (https://developer.android.com/reference/android/view/View.html#attr_android:focusableInTouchMode)
android: focusedByDefault (https://developer.android.com/reference/android/view/View.html#attr_android:focusedByDefault)
$\underline{and roid: force Has Overlapping Rendering} \ (https://developer.and roid.com/reference/and roid/view/View.html \# attr\_and roid: force Has Overlapping Rendering)$
android: foreground (https://developer.android.com/reference/android/view/View.html#attr_android:foreground)
<pre>android:foregroundGravity (https://developer.android.com/reference/android/view/View.html#attr_android:foregroundGravity)</pre>
<pre>android:foregroundTint (https://developer.android.com/reference/android/view/View.html#attr_android:foregroundTint)</pre>
<pre>android:foregroundTintMode (https://developer.android.com/reference/android/view/View.html#attr_android:foregroundTintMode)</pre>

android:hapticFeedbackEnabled (https://developer.android.com/reference/android/view/View.html#attr_android:hapticFeedbackEnabled)
android:id (https://developer.android.com/reference/android/view/View.html#attr_android:id)
<pre>android:importantForAccessibility (https://developer.android.com/reference/android/view/View.html#attr_android:importantForAccessibility)</pre>
<pre>android:importantForAutofill (https://developer.android.com/reference/android/view/View.html#attr_android:importantForAutofill)</pre>
<pre>android:isScrollContainer (https://developer.android.com/reference/android/view/View.html#attr_android:isScrollContainer)</pre>

<u>android:keepScreen0n</u> (https://developer.android.com/reference/android/view/View.html#attr_android:keepScreen0n)
android:keyboardNavigationCluster (https://developer.android.com/reference/android/view/View.html#attr_android:keyboardNavigationCluster
<pre>android:layerType (https://developer.android.com/reference/android/view/View.html#attr_android:layerType)</pre>
android:layoutDirection (https://developer.android.com/reference/android/view/View.html#attr_android:layoutDirection)
<pre>android:longClickable (https://developer.android.com/reference/android/view/View.html#attr_android:longClickable)</pre>
android:minHeight (https://developer.android.com/reference/android/view/View.html#attr_android:minHeight)
android:minWidth (https://developer.android.com/reference/android/view/View.html#attr_android:minWidth)
<pre>android:nextClusterForward (https://developer.android.com/reference/android/view/View.html#attr_android:nextClusterForward)</pre>
<pre>android:nextFocusDown (https://developer.android.com/reference/android/view/View.html#attr_android:nextFocusDown)</pre>





android:outlineAmbientShadowColor (https://developer.android.com/reference/android/view/View.html#attr_android:outlineAmbientShadowColor
<pre>android:outlineSpotShadowColor</pre> (https://developer.android.com/reference/android/view/View.html#attr_android:outlineSpotShadowColor)
android:padding (https://developer.android.com/reference/android/view/View.html#attr_android:padding)
<pre>android:paddingBottom (https://developer.android.com/reference/android/view/View.html#attr_android:paddingBottom)</pre>
<pre>android:paddingEnd (https://developer.android.com/reference/android/view/View.html#attr_android:paddingEnd)</pre>
<u>android:paddingHorizontal</u> (https://developer.android.com/reference/android/view/View.html#attr_android:paddingHorizontal)

android:paddingLeft (https://developer.android.com/reference/android/view/View.html#attr_android:paddingLeft)
<pre>android:paddingRight (https://developer.android.com/reference/android/view/View.html#attr_android:paddingRight)</pre>
<pre>android:paddingStart (https://developer.android.com/reference/android/view/View.html#attr_android:paddingStart)</pre>
android:paddingTop (https://developer.android.com/reference/android/view/View.html#attr_android:paddingTop)

android:paddingVertical (https://developer.android.com/reference/android/view/View.html#attr_android:paddingVertical)
android: requiresFadingEdge (https://developer.android.com/reference/android/view/View.html#attr_android:requiresFadingEdge)
<pre>android:rotation (https://developer.android.com/reference/android/view/View.html#attr_android:rotation)</pre>
<pre>android:rotationX (https://developer.android.com/reference/android/view/View.html#attr_android:rotationX)</pre>
<pre>android:rotationY (https://developer.android.com/reference/android/view/View.html#attr_android:rotationY)</pre>
<pre>android:saveEnabled (https://developer.android.com/reference/android/view/View.html#attr_android:saveEnabled)</pre>
<pre>android:scaleX (https://developer.android.com/reference/android/view/View.html#attr_android:scaleX)</pre>
<pre>android:scaleY (https://developer.android.com/reference/android/view/View.html#attr_android:scaleY)</pre>
android:scrollIndicators (https://developer.android.com/reference/android/view/View.html#attr_android:scrollIndicators)
<pre>android:scrollX (https://developer.android.com/reference/android/view/View.html#attr_android:scrollX)</pre>

<pre>android:scrollY (https://developer.android.com/reference/android/view/View.html#attr_android:scrollY)</pre>
<u>android:scrollbarAlwaysDrawHorizontalTrack</u> (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarAlways
android:scrollbarAlwaysDrawVerticalTrack (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarAlwaysDrawVerticalTrack
android:scrollbarDefaultDelayBeforeFade (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarDefaultDelayBeforeFade)
<pre>android:scrollbarFadeDuration (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarFadeDuration)</pre>
<pre>android:scrollbarSize (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarSize)</pre>
<pre>android:scrollbarStyle (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarStyle)</pre>
<pre>android:scrollbarThumbHorizontal (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarThumbHorizontal)</pre>
<pre>android:scrollbarThumbVertical (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarThumbVertical)</pre>
<pre>android:scrollbarTrackHorizontal (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarTrackHorizontal)</pre>

android:scrollbarTrackVertical (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbarTrackVertical)		
android:scrollbars (https://developer.android.com/reference/android/view/View.html#attr_android:scrollbars)		
android:soundEffectsEnabled (https://developer.android.com/reference/android/view/View.html#attr_android:soundEffectsEnabled)		
<pre>android:stateListAnimator</pre> (https://developer.android.com/reference/android/view/View.html#attr_android:stateListAnimator)		
android:tag (https://developer.android.com/reference/android/view/View.html#attr_android:tag)		
<pre>android:textAlignment (https://developer.android.com/reference/android/view/View.html#attr_android:textAlignment)</pre>		
<pre>android:textDirection (https://developer.android.com/reference/android/view/View.html#attr_android:textDirection)</pre>		
android: theme (https://developer.android.com/reference/android/view/View.html#attr_android:theme)		

<pre>android:tooltipText (https://developer.android.com/reference/android/view/View.html#attr_android:tooltipText)</pre>
<pre>android:transformPivotX (https://developer.android.com/reference/android/view/View.html#attr_android:transformPivotX)</pre>
<u>android:transformPivotY</u> (https://developer.android.com/reference/android/view/View.html#attr_android:transformPivotY)
<pre>android:transitionName (https://developer.android.com/reference/android/view/View.html#attr_android:transitionName)</pre>
<pre>android:translationX (https://developer.android.com/reference/android/view/View.html#attr_android:translationX)</pre>
<pre>android:translationY (https://developer.android.com/reference/android/view/View.html#attr_android:translationY)</pre>
<pre>android:translationZ (https://developer.android.com/reference/android/view/View.html#attr_android:translationZ)</pre>
<pre>android:visibility (https://developer.android.com/reference/android/view/View.html#attr_android:visibility)</pre>

# Inherited constants

From class <a href="mailto:android.view.View">android.view.View</a>. (https://developer.android.com/reference/android/view/View.html)

int

	(https://developer.android.com/reference/android/view/View.html#ACCES SIBILITY_LIVE_REGION_ASSERTIVE)
	Live region mode specifying that accessibility services should interrupt ongoing speech to immediately announce changes to this view.
int	ACCESSIBILITY_LIVE_REGION_NONE  (https://developer.android.com/reference/android/view/View.html#ACCESSI BILITY_LIVE_REGION_NONE)
	Live region mode specifying that accessibility services should not automatically announce changes to this view.
int	ACCESSIBILITY_LIVE_REGION_POLITE  (https://developer.android.com/reference/android/view/View.html#ACCESSI BILITY_LIVE_REGION_POLITE)
	Live region mode specifying that accessibility services should announce changes to this view.
int	AUTOFILL_FLAG_INCLUDE_NOT_IMPORTANT_VIEWS  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _FLAG_INCLUDE_NOT_IMPORTANT_VIEWS)
	Flag requesting you to add views that are marked as not important for autofill (see <a href="mailto:see1mportantForAutofill(int)">setImportantForAutofill(int)</a>
	(https://developer.android.com/reference/android/view/View.html#setImport antForAutofill(int))
	) to a <u>ViewStructure</u> (https://developer.android.com/reference/android/view/ViewStructure.html).
String (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_DATE  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_CREDIT_CARD_EXPIRATION_DATE)

	Hint indicating that this view can be autofilled with a credit card expiration date.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_DAY  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_CREDIT_CARD_EXPIRATION_DAY)
	Hint indicating that this view can be autofilled with a credit card expiration day.
String (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_MONTH  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_CREDIT_CARD_EXPIRATION_MONTH)
	Hint indicating that this view can be autofilled with a credit card expiration month.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_CREDIT_CARD_EXPIRATION_YEAR  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_CREDIT_CARD_EXPIRATION_YEAR)
	Hint indicating that this view can be autofilled with a credit card expiration year.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_CREDIT_CARD_NUMBER  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_CREDIT_CARD_NUMBER)
	Hint indicating that this view can be autofilled with a credit card number.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_CREDIT_CARD_SECURITY_CODE  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_CREDIT_CARD_SECURITY_CODE)

	Hint indicating that this view can be autofilled with a credit card security code.
String (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_EMAIL_ADDRESS  (https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_EMAIL_ADDRESS)
	Hint indicating that this view can be autofilled with an email address.
String (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_NAME (https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_NAME)
	Hint indicating that this view can be autofilled with a user's real name.
String (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_PASSWORD  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_PASSWORD)
	Hint indicating that this view can be autofilled with a password.
String (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_PHONE  (https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_PHONE)
	Hint indicating that this view can be autofilled with a phone number.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_POSTAL_ADDRESS  (https://developer.android.com/reference/android/view/View.html#AUTOFILL_HINT_POSTAL_ADDRESS)
	Hint indicating that this view can be autofilled with a postal address.
<u>String</u> (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_POSTAL_CODE

	_HINT_POSTAL_CODE)
	Hint indicating that this view can be autofilled with a postal code.
String (https://developer.android.com/reference/java/lang/String.html)	AUTOFILL_HINT_USERNAME (https://developer.android.com/reference/android/view/View.html#AUTOFILL _HINT_USERNAME)
	Hint indicating that this view can be autofilled with a username.
int	AUTOFILL_TYPE_DATE (https://developer.android.com/reference/android/view/View.html#AUTOFILL _TYPE_DATE)
	Autofill type for a field that contains a date, which is represented by a long representing the number of milliseconds since the standard base time known as "the epoch", namely January 1, 1970, 00:00:00 GMT (see <a href="Date.getTime()">Date.getTime()</a> . (https://developer.android.com/reference/java/util/Date.html#getTime()).
int	AUTOFILL_TYPE_LIST  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _TYPE_LIST)
	Autofill type for a selection list field, which is filled by an <b>int</b> representing the element index inside the list (starting at 0).
int	AUTOFILL_TYPE_NONE (https://developer.android.com/reference/android/view/View.html#AUTOFILL_TYPE_NONE)
	Autofill type for views that cannot be autofilled.

(https://developer.android.com/reference/android/view/View.html#AUTOFILL

int	<u>AUTOFILL_TYPE_TEXT</u> (https://developer.android.com/reference/android/view/View.html#AUTOFILL_TYPE_TEXT)
	Autofill type for a text field, which is filled by a <a href="CharSequence">CharSequence</a> (https://developer.android.com/reference/java/lang/CharSequence.html).
int	AUTOFILL_TYPE_TOGGLE  (https://developer.android.com/reference/android/view/View.html#AUTOFILL _TYPE_TOGGLE)
	Autofill type for a togglable field, which is filled by a <b>boolean</b> .
int	<pre>DRAG_FLAG_GLOBAL   (https://developer.android.com/reference/android/view/View.html#DRAG_FL AG_GLOBAL)</pre>
	Flag indicating that a drag can cross window boundaries.
int	DRAG_FLAG_GLOBAL_PERSISTABLE_URI_PERMISSION  (https://developer.android.com/reference/android/view/View.html#DRAG_FL AG_GLOBAL_PERSISTABLE_URI_PERMISSION)  When this flag is used with DRAG_FLAG_GLOBAL_URI_READ  (https://developer.android.com/reference/android/view/View.html#DRAG_FL AG_GLOBAL_URI_READ) and/or DRAG_FLAG_GLOBAL_URI_WRITE  (https://developer.android.com/reference/android/view/View.html#DRAG_FL AG_GLOBAL_URI_WRITE) , the URI permission grant can be persisted across device reboots until explicitly revoked with Context.revokeUriPermission(Uri, int)

	(https://developer.android.com/reference/android/content/Context.html#revokeUriPermission(android.net.Uri,%20int)) Context.revokeUriPermission}.
int	<u>DRAG_FLAG_GLOBAL_PREFIX_URI_PERMISSION</u> (https://developer.android.com/reference/android/view/View.html#DRAG_FL AG_GLOBAL_PREFIX_URI_PERMISSION)
	When this flag is used with <a href="mailto:DRAG_FLAG_GLOBAL_URI_READ">DRAG_FLAG_GLOBAL_URI_READ</a> (https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_URI_WRITE and/or <a href="mailto:DRAG_FLAG_GLOBAL_URI_WRITE">DRAG_FLAG_GLOBAL_URI_WRITE</a> (https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_URI_WRITE) , the URI permission grant applies to any URI that is a prefix match against the original granted URI.
int	<u>DRAG_FLAG_GLOBAL_URI_READ</u> (https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GLOBAL_URI_READ)
	When this flag is used with <a href="mailto:DRAG_FLAG_GL0BAL">DRAG_FLAG_GL0BAL</a> (https://developer.android.com/reference/android/view/View.html#DRAG_FLAG_GL0BAL) , the drag recipient will be able to request read access to the content URI(s) contained in the <a href="mailto:ClipData">ClipData</a> (https://developer.android.com/reference/android/content/ClipData.html) object.
int	<u>DRAG_FLAG_GLOBAL_URI_WRITE</u> (https://developer.android.com/reference/android/view/View.html#DRAG_FL AG_GLOBAL_URI_WRITE)

int int

When this flag is used with <a href="mailto:DRAG\_FLAG\_GLOBAL">DRAG\_FLAG\_GLOBAL</a>

(https://developer.android.com/reference/android/view/View.html#DRAG\_FL AG\_GLOBAL)

, the drag recipient will be able to request write access to the content URI(s) contained in the ClipData

(https://developer.android.com/reference/android/content/ClipData.html) object.

# DRAG\_FLAG\_OPAQUE

(https://developer.android.com/reference/android/view/View.html#DRAG\_FL AG\_OPAQUE)

Flag indicating that the drag shadow will be opaque.

### DRAWING\_CACHE\_QUALITY\_AUTO

(https://developer.android.com/reference/android/view/View.html#DRAWING \_CACHE\_QUALITY\_AUTO)

This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, <a href="mailto:setLayerType(int, Paint">setLayerType(int, Paint)</a>. (https://developer.android.com/reference/android/view/View.html#setLayerType(int,%20android.graphics.Paint))

handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a <u>Canvas</u>

(https://developer.android.com/reference/android/graphics/Canvas.html) from either a <u>Bitmap</u>

(https://developer.android.com/reference/android/graphics/Bitmap.html) or Picture

(https://developer.android.com/reference/android/graphics/Picture.html) and call <u>draw(Canvas)</u>

(https://developer.android.com/reference/android/view/View.html#draw(and roid.graphics.Canvas))

on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as <a href="Config.HARDWARE">Config.HARDWARE</a>

(https://developer.android.com/reference/android/graphics/Bitmap.Config.ht ml#HARDWARE)

bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the <a href="PixelCopy">PixelCopy</a>

(https://developer.android.com/reference/android/view/PixelCopy.html) *API is recommended.* 

### DRAWING\_CACHE\_QUALITY\_HIGH

(https://developer.android.com/reference/android/view/View.html#DRAWING \_CACHE\_QUALITY\_HIGH)

This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, <a href="mailto:setLayerType(int, Paint)">setLayerType(int, Paint)</a>.

(https://developer.android.com/reference/android/view/View.html#setLayerType(int,%20android.graphics.Paint))

handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a Canvas

(https://developer.android.com/reference/android/graphics/Canvas.html) from either a <u>Bitmap</u>

(https://developer.android.com/reference/android/graphics/Bitmap.html) or

int

#### <u>Picture</u>

(https://developer.android.com/reference/android/graphics/Picture.html) and call <u>draw(Canvas)</u>

(https://developer.android.com/reference/android/view/View.html#draw(and roid.graphics.Canvas))

on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as

# Config. HARDWARE

(https://developer.android.com/reference/android/graphics/Bitmap.Config.ht ml#HARDWARE)

bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the <u>PixelCopy</u>

(https://developer.android.com/reference/android/view/PixelCopy.html) *API* is recommended.

### DRAWING\_CACHE\_QUALITY\_LOW

(https://developer.android.com/reference/android/view/View.html#DRAWING \_CACHE\_QUALITY\_LOW)

This constant was deprecated in API level 28. The view drawing cache was largely made obsolete with the introduction of hardware-accelerated rendering in API 11. With hardware-acceleration, intermediate cache layers are largely unnecessary and can easily result in a net loss in performance due to the cost of creating and updating the layer. In the rare cases where caching layers are useful, such as for alpha animations, <a href="mailto:setLayerType(int, Paint)">setLayerType(int, Paint)</a>.

(https://developer.android.com/reference/android/view/View.html#setLayerType(int,%20android.graphics.Paint))

handles this with hardware rendering. For software-rendered snapshots of a small part of the View hierarchy or individual Views it is recommended to create a <u>Canvas</u>

(https://developer.android.com/reference/android/graphics/Canvas.html) from either a <u>Bitmap</u>

int

(https://developer.android.com/reference/android/graphics/Bitmap.html) or **Picture** (https://developer.android.com/reference/android/graphics/Picture.html) and call draw(Canvas) (https://developer.android.com/reference/android/view/View.html#draw(and roid.graphics.Canvas)) on the View. However these software-rendered usages are discouraged and have compatibility issues with hardware-only rendering features such as Config.HARDWARE (https://developer.android.com/reference/android/graphics/Bitmap.Config.ht ml#HARDWARE) bitmaps, real-time shadows, and outline clipping. For screenshots of the UI for feedback reports or unit testing the PixelCopy (https://developer.android.com/reference/android/view/PixelCopy.html) API is recommended. FIND\_VIEWS\_WITH\_CONTENT\_DESCRIPTION int (https://developer.android.com/reference/android/view/View.html#FIND\_VIE WS\_WITH\_CONTENT\_DESCRIPTION) Find find views that contain the specified content description. int FIND\_VIEWS\_WITH\_TEXT (https://developer.android.com/reference/android/view/View.html#FIND\_VIE WS\_WITH\_TEXT) Find views that render the specified text. **FOCUSABLE** int (https://developer.android.com/reference/android/view/View.html#FOCUSAB LE) This view wants keystrokes.

int	FOCUSABLES_ALL (https://developer.android.com/reference/android/view/View.html#FOCUSAB LES_ALL)
	View flag indicating whether <a href="mailto:addFocusables">addFocusables</a> (ArrayList, int, int) (https://developer.android.com/reference/android/view/View.html#addFocusables(java.util.ArrayList <android.view.view>,%20int,%20int)) should add all focusable Views regardless if they are focusable in touch mode.</android.view.view>
int	FOCUSABLES_TOUCH_MODE  (https://developer.android.com/reference/android/view/View.html#FOCUSAB LES_TOUCH_MODE)
	View flag indicating whether <a href="mailto:addFocusables">addFocusables</a> (ArrayList, int, int) (https://developer.android.com/reference/android/view/View.html#addFocus ables(java.util.ArrayList <android.view.view>,%20int,%20int)) should add only Views focusable in touch mode.</android.view.view>
int	FOCUSABLE_AUTO  (https://developer.android.com/reference/android/view/View.html#FOCUSAB LE_AUTO)
	This view determines focusability automatically.
int	FOCUS_BACKWARD  (https://developer.android.com/reference/android/view/View.html#FOCUS_BACKWARD)
	Use with <a href="mailto:focusSearch(int">focusSearch(int)</a> (https://developer.android.com/reference/android/view/View.html#focusSearch(int)) .

int	FOCUS_DOWN  (https://developer.android.com/reference/android/view/View.html#FOCUS_DOWN)  Use with focusSearch(int)  (https://developer.android.com/reference/android/view/View.html#focusSearch(int))
int	FOCUS_FORWARD  (https://developer.android.com/reference/android/view/View.html#FOCUS_FORWARD)  Use with focusSearch(int) (https://developer.android.com/reference/android/view/View.html#focusSearch(int)) .
int	FOCUS_LEFT  (https://developer.android.com/reference/android/view/View.html#FOCUS_L EFT)  Use with focusSearch(int) (https://developer.android.com/reference/android/view/View.html#focusSearch(int)) .
int	FOCUS_RIGHT  (https://developer.android.com/reference/android/view/View.html#FOCUS_RIGHT)

	Use with <a href="mailto:focusSearch(int">focusSearch(int)</a> (https://developer.android.com/reference/android/view/View.html#focusSearch(int)) .
int	FOCUS_UP (https://developer.android.com/reference/android/view/View.html#FOCUS_UP)
	Use with <a href="mailto:focusSearch(int">focusSearch(int)</a> (https://developer.android.com/reference/android/view/View.html#focusSearch(int)) .
int	GONE (https://developer.android.com/reference/android/view/View.html#GONE)
	This view is invisible, and it doesn't take any space for layout purposes.
int	HAPTIC_FEEDBACK_ENABLED  (https://developer.android.com/reference/android/view/View.html#HAPTIC_FEEDBACK_ENABLED)
	View flag indicating whether this view should have haptic feedback enabled for events such as long presses.
int	IMPORTANT_FOR_ACCESSIBILITY_AUTO  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_ACCESSIBILITY_AUTO)
	Automatically determine whether a view is important for accessibility.
int	<pre>IMPORTANT_FOR_ACCESSIBILITY_NO</pre>

	(https://developer.android.com/reference/android/view/View.html#IMPORTANT_FOR_ACCESSIBILITY_NO)
	The view is not important for accessibility.
int	IMPORTANT_FOR_ACCESSIBILITY_NO_HIDE_DESCENDANTS  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_ACCESSIBILITY_NO_HIDE_DESCENDANTS)  The view is not important for accessibility, nor are any of its descendant views.
int	IMPORTANT_FOR_ACCESSIBILITY_YES  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_ACCESSIBILITY_YES)  The view is important for accessibility.
int	IMPORTANT_FOR_AUTOFILL_AUTO  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_AUTOFILL_AUTO)  Automatically determine whether a view is important for autofill.
int	IMPORTANT_FOR_AUTOFILL_NO  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_AUTOFILL_NO)  The view is not important for autofill, but its children (if any) will be traversed.
int	IMPORTANT_FOR_AUTOFILL_NO_EXCLUDE_DESCENDANTS  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_AUTOFILL_NO_EXCLUDE_DESCENDANTS)

	The view is not important for autofill, and its children (if any) will not be traversed.
int	IMPORTANT_FOR_AUTOFILL_YES  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_AUTOFILL_YES)
	The view is important for autofill, and its children (if any) will be traversed.
int	IMPORTANT_FOR_AUTOFILL_YES_EXCLUDE_DESCENDANTS  (https://developer.android.com/reference/android/view/View.html#IMPORTA NT_FOR_AUTOFILL_YES_EXCLUDE_DESCENDANTS)
	The view is important for autofill, but its children (if any) will not be traversed.
int	INVISIBLE (https://developer.android.com/reference/android/view/View.html#INVISIBLE)
	This view is invisible, but it still takes up space for layout purposes.
int	KEEP_SCREEN_ON (https://developer.android.com/reference/android/view/View.html#KEEP_SC REEN_ON)
	View flag indicating that the screen should remain on while the window containing this view is visible to the user.
int	<u>LAYER_TYPE_HARDWARE</u> (https://developer.android.com/reference/android/view/View.html#LAYER_T YPE_HARDWARE)
	Indicates that the view has a hardware layer.

int	<u>LAYER_TYPE_NONE</u> (https://developer.android.com/reference/android/view/View.html#LAYER_T YPE_NONE)
	Indicates that the view does not have a layer.
int	<u>LAYER_TYPE_SOFTWARE</u> (https://developer.android.com/reference/android/view/View.html#LAYER_T YPE_SOFTWARE)
	Indicates that the view has a software layer.
int	<u>LAYOUT_DIRECTION_INHERIT</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_INHERIT)
	Horizontal layout direction of this view is inherited from its parent.
int	<u>LAYOUT_DIRECTION_LOCALE</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_LOCALE)
	Horizontal layout direction of this view is from deduced from the default language script for the locale.
int	<u>LAYOUT_DIRECTION_LTR</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_LTR)
	Horizontal layout direction of this view is from Left to Right.
int	<u>LAYOUT_DIRECTION_RTL</u> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_RTL)

Horizontal layout direction of this view is from Right to Left.

#### int

# MEASURED\_HEIGHT\_STATE\_SHIFT

(https://developer.android.com/reference/android/view/View.html#MEASUR ED\_HEIGHT\_STATE\_SHIFT)

### Bit shift of MEASURED\_STATE\_MASK

(https://developer.android.com/reference/android/view/View.html#MEASUR ED\_STATE\_MASK)

to get to the height bits for functions that combine both width and height into a single int, such as <u>getMeasuredState()</u>

(https://developer.android.com/reference/android/view/View.html#getMeasuredState())

and the childState argument of <a href="resolveSizeAndState(int, int, int)">resolveSizeAndState(int, int, int)</a> (https://developer.android.com/reference/android/view/View.html#resolveSizeAndState(int,%20int,%20int))

#### •

#### int

## MEASURED\_SIZE\_MASK

(https://developer.android.com/reference/android/view/View.html#MEASUR ED\_SIZE\_MASK)

# Bits of **getMeasuredWidthAndState**()

(https://developer.android.com/reference/android/view/View.html#getMeasuredWidthAndState())

# and getMeasuredWidthAndState()

(https://developer.android.com/reference/android/view/View.html #getMeasuredWidthAndState())

that provide the actual measured size.

#### int

## MEASURED\_STATE\_MASK

(https://developer.android.com/reference/android/view/View.html#MEASUR ED\_STATE\_MASK)

	Bits of getMeasuredWidthAndState() (https://developer.android.com/reference/android/view/View.html#getMeasuredWidthAndState()) and getMeasuredWidthAndState() (https://developer.android.com/reference/android/view/View.html#getMeasuredWidthAndState()) that provide the additional state bits.
int	MEASURED_STATE_TOO_SMALL  (https://developer.android.com/reference/android/view/View.html#MEASUR ED_STATE_TOO_SMALL)  Bit of getMeasuredWidthAndState()  (https://developer.android.com/reference/android/view/View.html#getMeasuredWidthAndState()) and getMeasuredWidthAndState()  (https://developer.android.com/reference/android/view/View.html#getMeasuredWidthAndState()) that indicates the measured size is smaller that the space the view would like to have.
int	NOT_FOCUSABLE  (https://developer.android.com/reference/android/view/View.html#NOT_FOCUSABLE)  This view does not want keystrokes.
int	NO_ID  (https://developer.android.com/reference/android/view/View.html#NO_ID)  Used to mark a View that has no ID.
int	OVER_SCROLL_ALWAYS

	(https://developer.android.com/reference/android/view/View.html#OVER_SC ROLL_ALWAYS)
	Always allow a user to over-scroll this view, provided it is a view that can scroll.
int	OVER_SCROLL_IF_CONTENT_SCROLLS  (https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_IF_CONTENT_SCROLLS)
	Allow a user to over-scroll this view only if the content is large enough to meaningfully scroll, provided it is a view that can scroll.
int	OVER_SCROLL_NEVER  (https://developer.android.com/reference/android/view/View.html#OVER_SCROLL_NEVER)
	Never allow a user to over-scroll this view.
int	SCREEN_STATE_OFF  (https://developer.android.com/reference/android/view/View.html#SCREEN_STATE_OFF)
	Indicates that the screen has changed state and is now off.
int	SCREEN_STATE_ON  (https://developer.android.com/reference/android/view/View.html#SCREEN_STATE_ON)
	Indicates that the screen has changed state and is now on.
int	SCROLLBARS_INSIDE_INSET  (https://developer.android.com/reference/android/view/View.html#SCROLLB ARS_INSIDE_INSET)

	The scrollbar style to display the scrollbars inside the padded area, increasing the padding of the view.
int	SCROLLBARS_INSIDE_OVERLAY  (https://developer.android.com/reference/android/view/View.html#SCROLLB ARS_INSIDE_OVERLAY)
	The scrollbar style to display the scrollbars inside the content area, without increasing the padding.
int	SCROLLBARS_OUTSIDE_INSET  (https://developer.android.com/reference/android/view/View.html#SCROLLB ARS_OUTSIDE_INSET)
	The scrollbar style to display the scrollbars at the edge of the view, increasing the padding of the view.
int	SCROLLBARS_OUTSIDE_OVERLAY  (https://developer.android.com/reference/android/view/View.html#SCROLLB ARS_OUTSIDE_OVERLAY)
	The scrollbar style to display the scrollbars at the edge of the view, without increasing the padding.
int	SCROLLBAR_POSITION_DEFAULT  (https://developer.android.com/reference/android/view/View.html#SCROLLB AR_POSITION_DEFAULT)
	Position the scroll bar at the default position as determined by the system.
int	<u>SCROLLBAR_POSITION_LEFT</u> (https://developer.android.com/reference/android/view/View.html#SCROLLBAR_POSITION_LEFT)

	Position the scroll bar along the left edge.
int	SCROLLBAR_POSITION_RIGHT  (https://developer.android.com/reference/android/view/View.html#SCROLLBAR_POSITION_RIGHT)
	Position the scroll bar along the right edge.
int	SCROLL_AXIS_HORIZONTAL  (https://developer.android.com/reference/android/view/View.html#SCROLL_AXIS_HORIZONTAL)
	Indicates scrolling along the horizontal axis.
int	SCROLL_AXIS_NONE  (https://developer.android.com/reference/android/view/View.html#SCROLL_AXIS_NONE)
	Indicates no axis of view scrolling.
int	SCROLL_AXIS_VERTICAL  (https://developer.android.com/reference/android/view/View.html#SCROLL_AXIS_VERTICAL)
	Indicates scrolling along the vertical axis.
int	SCROLL_INDICATOR_BOTTOM  (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_BOTTOM)
	Scroll indicator direction for the bottom edge of the view.
int	SCROLL_INDICATOR_END

	(https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_END)
	Scroll indicator direction for the ending edge of the view.
int	SCROLL_INDICATOR_LEFT  (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_LEFT)
	Scroll indicator direction for the left edge of the view.
int	SCROLL_INDICATOR_RIGHT  (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_RIGHT)
	Scroll indicator direction for the right edge of the view.
int	SCROLL_INDICATOR_START  (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_START)
	Scroll indicator direction for the starting edge of the view.
int	SCROLL_INDICATOR_TOP  (https://developer.android.com/reference/android/view/View.html#SCROLL_INDICATOR_TOP)
	Scroll indicator direction for the top edge of the view.
int	SOUND_EFFECTS_ENABLED (https://developer.android.com/reference/android/view/View.html#SOUND_EFFECTS_ENABLED)

	View flag indicating whether this view should have sound effects enabled for events such as clicking and touching.
int	STATUS_BAR_HIDDEN  (https://developer.android.com/reference/android/view/View.html#STATUS_BAR_HIDDEN)
	This constant was deprecated in API level 14. Use  SYSTEM_UI_FLAG_LOW_PROFILE  (https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_LOW_PROFILE)  instead.
int	STATUS_BAR_VISIBLE  (https://developer.android.com/reference/android/view/View.html#STATUS_BAR_VISIBLE)
	This constant was deprecated in API level 14. Use <a href="mailto:SYSTEM_UI_FLAG_VISIBLE">SYSTEM_UI_FLAG_VISIBLE</a> (https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_VISIBLE) instead.
int	SYSTEM_UI_FLAG_FULLSCREEN  (https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_FULLSCREEN)
	Flag for <u>setSystemUiVisibility(int)</u> (https://developer.android.com/reference/android/view/View.html#setSyste mUiVisibility(int)) : View has requested to go into the normal fullscreen mode so that its content can take over the screen while still allowing the user to interact with the application.

int

### SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION)

# Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int))

: View has requested that the system navigation be temporarily hidden.

int

### SYSTEM\_UI\_FLAG\_IMMERSIVE

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_ UI\_FLAG\_IMMERSIVE)

# Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSyste mUiVisibility(int))

: View would like to remain interactive when hiding the navigation bar with **SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION** 

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION)

.

int

## SYSTEM\_UI\_FLAG\_IMMERSIVE\_STICKY

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_ UI\_FLAG\_IMMERSIVE\_STICKY)

## Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int))

: View would like to remain interactive when hiding the status bar with

# SYSTEM\_UI\_FLAG\_FULLSCREEN

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_FULLSCREEN)

and/or hiding the navigation bar with <a href="mailto:SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION">SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION</a> (https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION)

.

#### int

### SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_LAYOUT\_FULLSCREEN)

# Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int))

: View would like its window to be laid out as if it has requested

## SYSTEM\_UI\_FLAG\_FULLSCREEN

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_FULLSCREEN)

, even if it currently hasn't.

#### int

# SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_LAYOUT\_HIDE\_NAVIGATION)

# Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSyste mUiVisibility(int))

: View would like its window to be laid out as if it has requested

# SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_HIDE\_NAVIGATION)

, even if it currently hasn't.

# SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_LAYOUT\_STABLE)

# Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int))

: When using other layout flags, we would like a stable view of the content insets given to <a href="fitSystemWindows(Rect">fitSystemWindows(Rect)</a>

(https://developer.android.com/reference/android/view/View.html#fitSystem Windows(android.graphics.Rect))

.

#### int

# SYSTEM\_UI\_FLAG\_LIGHT\_NAVIGATION\_BAR

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_LIGHT\_NAVIGATION\_BAR)

# Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int))

: Requests the navigation bar to draw in a mode that is compatible with light navigation bar backgrounds.

#### int

# SYSTEM\_UI\_FLAG\_LIGHT\_STATUS\_BAR

(https://developer.android.com/reference/android/view/View.html#SYSTEM\_UI\_FLAG\_LIGHT\_STATUS\_BAR)

# Flag for setSystemUiVisibility(int)

(https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int))

: Requests the status bar to draw in a mode that is compatible with light status bar backgrounds.

# SYSTEM\_UI\_FLAG\_LOW\_PROFILE

#### int

	(https://developer.android.com/reference/android/view/View.html#SYSTEM_ UI_FLAG_LOW_PROFILE)
	Flag for <a href="mailto:setSystemUiVisibility">setSystemUiVisibility</a> (int) (https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int)) : View has requested the system UI to enter an unobtrusive "low profile" mode.
int	SYSTEM_UI_FLAG_VISIBLE  (https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_FLAG_VISIBLE)
	Special constant for <a href="mailto:setSystemUiVisibility(int">setSystemUiVisibility(int)</a> (https://developer.android.com/reference/android/view/View.html#setSystemUiVisibility(int)) : View has requested the system UI (status bar) to be visible (the default).
int	SYSTEM_UI_LAYOUT_FLAGS  (https://developer.android.com/reference/android/view/View.html#SYSTEM_UI_LAYOUT_FLAGS)
	Flags that can impact the layout in relation to system UI.
int	TEXT_ALIGNMENT_CENTER  (https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_CENTER)
	Center the paragraph, e.g.
int	TEXT_ALIGNMENT_GRAVITY  (https://developer.android.com/reference/android/view/View.html#TEXT_ALIGNMENT_GRAVITY)
	Default for the root view.

int	TEXT_ALIGNMENT_INHERIT  (https://developer.android.com/reference/android/view/View.html#TEXT_ALGNMENT_INHERIT)
	Default text alignment.
int	TEXT_ALIGNMENT_TEXT_END  (https://developer.android.com/reference/android/view/View.html#TEXT_ALGNMENT_TEXT_END)
	Align to the end of the paragraph, e.g.
int	TEXT_ALIGNMENT_TEXT_START  (https://developer.android.com/reference/android/view/View.html#TEXT_ALGNMENT_TEXT_START)
	Align to the start of the paragraph, e.g.
int	TEXT_ALIGNMENT_VIEW_END  (https://developer.android.com/reference/android/view/View.html#TEXT_ALGNMENT_VIEW_END)
	Align to the end of the view, which is ALIGN_RIGHT if the view's resolved layoutDirection is LTR, and ALIGN_LEFT otherwise.
int	TEXT_ALIGNMENT_VIEW_START  (https://developer.android.com/reference/android/view/View.html#TEXT_ALGNMENT_VIEW_START)
	Align to the start of the view, which is ALIGN_LEFT if the view's resolved layoutDirection is LTR, and ALIGN_RIGHT otherwise.
int	TEXT_DIRECTION_ANY_RTL

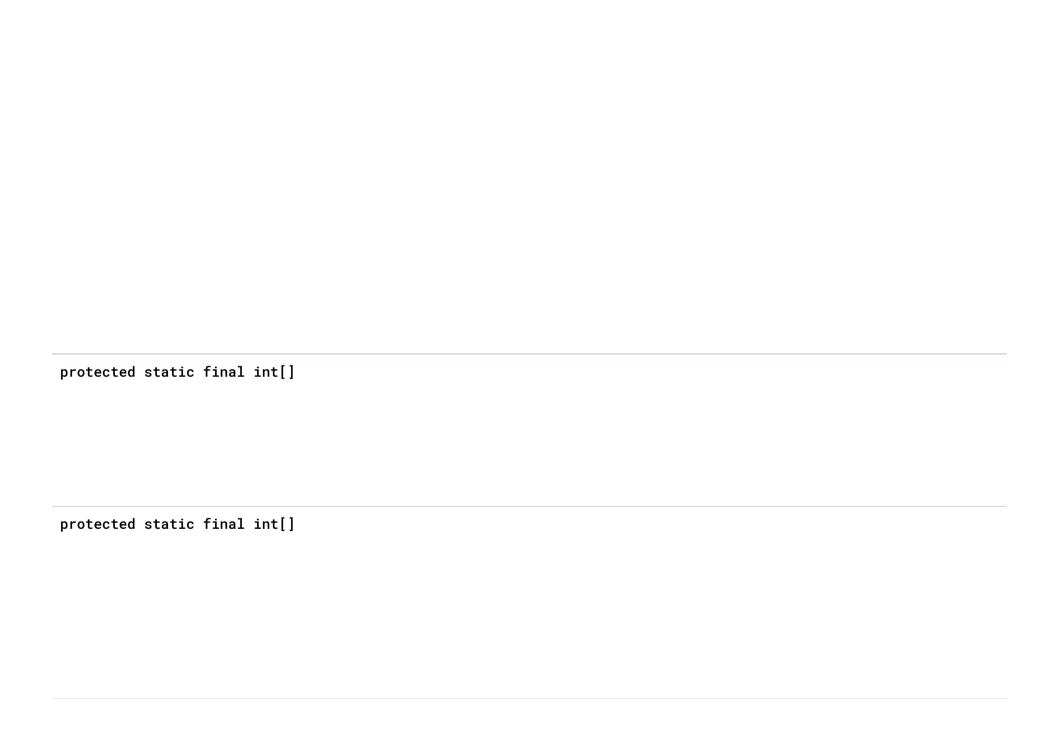
	(https://developer.android.com/reference/android/view/View.html#TEXT_DIR ECTION_ANY_RTL)
	Text direction is using "any-RTL" algorithm.
int	TEXT_DIRECTION_FIRST_STRONG  (https://developer.android.com/reference/android/view/View.html#TEXT_DIR ECTION_FIRST_STRONG)
	Text direction is using "first strong algorithm".
int	TEXT_DIRECTION_FIRST_STRONG_LTR  (https://developer.android.com/reference/android/view/View.html#TEXT_DIR ECTION_FIRST_STRONG_LTR)
	Text direction is using "first strong algorithm".
int	TEXT_DIRECTION_FIRST_STRONG_RTL (https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_FIRST_STRONG_RTL)
	Text direction is using "first strong algorithm".
int	<u>TEXT_DIRECTION_INHERIT</u> (https://developer.android.com/reference/android/view/View.html#TEXT_DIR ECTION_INHERIT)
	Text direction is inherited through <u>ViewGroup</u> (https://developer.android.com/reference/android/view/ViewGroup.html)
int	TEXT_DIRECTION_LOCALE  (https://developer.android.com/reference/android/view/View.html#TEXT_DIR ECTION_LOCALE)

	Text direction is coming from the system Locale.
int	TEXT_DIRECTION_LTR  (https://developer.android.com/reference/android/view/View.html#TEXT_DIR ECTION_LTR)
	Text direction is forced to LTR.
int	TEXT_DIRECTION_RTL (https://developer.android.com/reference/android/view/View.html#TEXT_DIRECTION_RTL)
	Text direction is forced to RTL.
String (https://developer.android.com/reference/java/lang/String.html)	VIEW_LOG_TAG  (https://developer.android.com/reference/android/view/View.html#VIEW_LOG_TAG)
	The logging tag used by this class with android.util.Log.
int	VISIBLE (https://developer.android.com/reference/android/view/View.html#VISIBLE)
	This view is visible.

# Inherited fields

From class <a href="mailto:android.view.View">android.view.View</a>. (https://developer.android.com/reference/android/view/View.html)

public static final Property (https://developer.android.com/reference/android/util/Property.html)
View (https://developer.android.com/reference/java/lang/Float.html)>







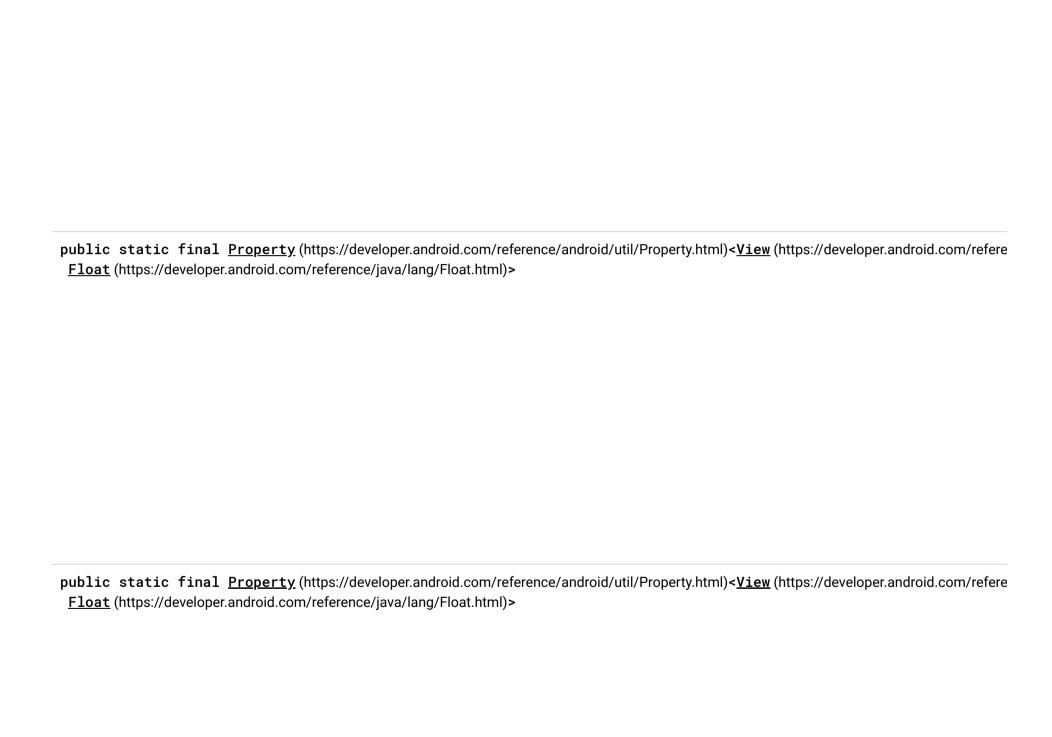
protected static final int[] protected static final int[]

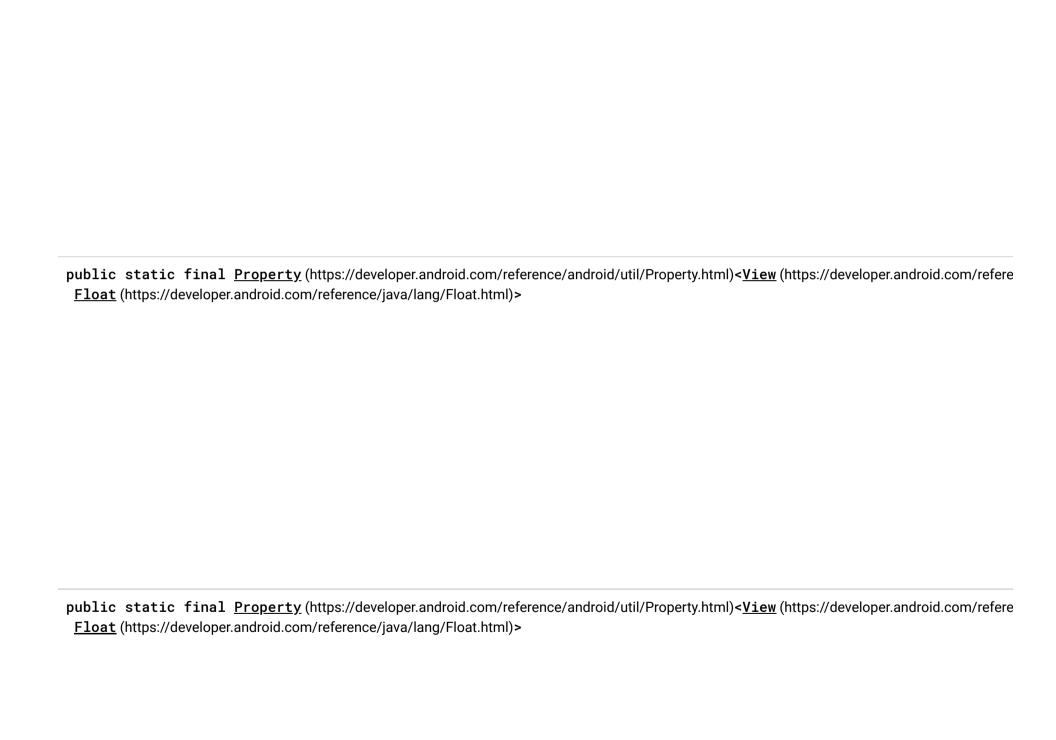


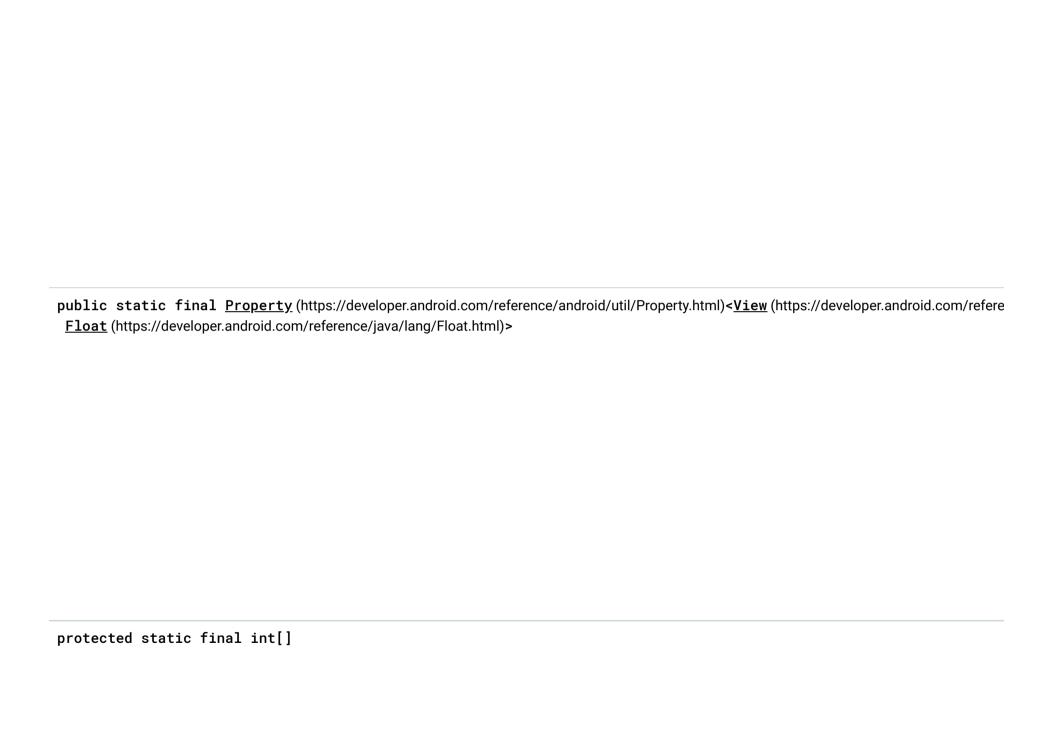


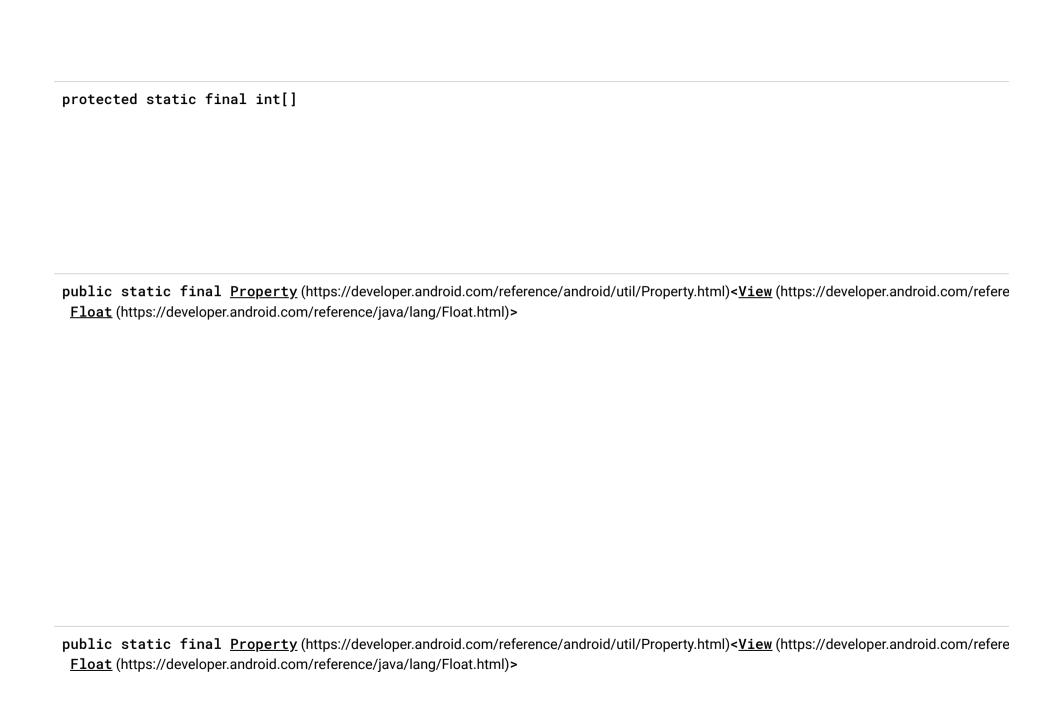


















### **Public constructors**

<u>VideoView</u> (https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context)) (<u>Context</u> (https://developer.android.com/reference/android/content/Context.html) <u>context</u>)

<u>VideoView</u> (https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context,%20android.util.AttributeSet))
(<u>Context</u> (https://developer.android.com/reference/android/content/Context.html) context, <u>AttributeSet</u>
(https://developer.android.com/reference/android/util/AttributeSet.html) attrs)

### VideoView

(https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context,%20android.util.AttributeSet,%20int))
(Context (https://developer.android.com/reference/android/content/Context.html) context, AttributeSet
(https://developer.android.com/reference/android/util/AttributeSet.html) attrs, int defStyleAttr)

#### <u>VideoView</u>

(https://developer.android.com/reference/android/widget/VideoView.html#VideoView(android.content.Context,%20android.util.AttributeSet,%20int,%20int)) (Context (https://developer.android.com/reference/android/content/Context.html) context, AttributeSet (https://developer.android.com/reference/android/util/AttributeSet.html) attrs, int defStyleAttr, int defStyleRes)

## **Public methods**

void	addSubtitleSource
	(https://developer.android.com/reference/android/widget/VideoView.html#addSubtitleSource(java.io.InputStream,%20android.media.MediaFormat))
	( <u>InputStream</u> (https://developer.android.com/reference/java/io/InputStream.html) is, <u>MediaFormat</u> (https://developer.android.com/reference/android/media/MediaFormat.html) format)
	Adds an external subtitle source file (from the provided input stream.) Note that a single external subtitle source may contain multiple or no supported tracks in it.

boolean	<pre>canPause (https://developer.android.com/reference/android/widget/VideoView.html#canPause())()</pre>
boolean	$\underline{\textbf{canSeekBackward}} \ (\text{https://developer.android.com/reference/android/widget/VideoView.html\#canSeekBackward()) (\ \textbf{)} \\$
boolean	<pre>canSeekForward (https://developer.android.com/reference/android/widget/VideoView.html#canSeekForward())()</pre>
void	<u>draw</u> (https://developer.android.com/reference/android/widget/VideoView.html#draw(android.graphics.Canvas)) ( <u>Canvas</u> (https://developer.android.com/reference/android/graphics/Canvas.html) <u>canvas</u> )
	Manually render this view (and all of its children) to the given Canvas.
ference/java/lang/CharSequence	getAccessibilityClassName e (https://developer.android.com/reference/android/widget/VideoView.html#getAccessibilityClassName())() . Return the class name of this object to be used for accessibility purposes.
html)	
int	<pre>getAudioSessionId (https://developer.android.com/reference/android/widget/VideoView.html#getAudioSessionId())()</pre> Get the audio session id for the player used by this VideoView.
int	<pre>getBufferPercentage (https://developer.android.com/reference/android/widget/VideoView.html#getBufferPercentage())()</pre>
int	$\underline{\texttt{getCurrentPosition}} \ (\text{https://developer.android.com/reference/android/widget/VideoView.html} \# getCurrentPosition()) \\ (\ )$
int	$\underline{\texttt{getDuration}} \ (\texttt{https://developer.android.com/reference/android/widget/VideoView.html\#getDuration())()}$
boolean	<u>isPlaying</u> (https://developer.android.com/reference/android/widget/VideoView.html#isPlaying())()
boolean	<pre>onKeyDown (https://developer.android.com/reference/android/widget/VideoView.html#onKeyDown(int,%20android.view.KeyEvent)) (int keyCode, KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) event)</pre>

	Default implementation of KeyEvent.Callback.onKeyDown()
	(https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int,%20android.view.KeyEvent))
	: perform press of the view when <b>KeyEvent.KEYCODE_DPAD_CENTER</b>
	(https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_DPAD_CENTER) or
	KeyEvent.KEYCODE_ENTER (https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE_ENTER) is
	released, if the view is enabled and clickable.
boolean	<u>onTouchEvent</u>
	(https://developer.android.com/reference/android/widget/VideoView.html#onTouchEvent(android.view.MotionEvent))
	( <u>MotionEvent</u> (https://developer.android.com/reference/android/view/MotionEvent.html) ev)
	Implement this method to handle touch screen motion events.
boolean	<u>onTrackballEvent</u>
	(https://developer.android.com/reference/android/widget/VideoView.html#onTrackballEvent(android.view.MotionEvent))
	( <u>MotionEvent</u> (https://developer.android.com/reference/android/view/MotionEvent.html) ev)
	Implement this method to handle trackball motion events.
void	<u>pause</u> (https://developer.android.com/reference/android/widget/VideoView.html#pause())()
int	<u>resolveAdjustedSize</u>
	(https://developer.android.com/reference/android/widget/VideoView.html#resolveAdjustedSize(int,%20int))(int
	desiredSize, int measureSpec)
void	<u>resume</u> (https://developer.android.com/reference/android/widget/VideoView.html#resume())()
void	<pre>seekTo (https://developer.android.com/reference/android/widget/VideoView.html#seekTo(int))(int msec)</pre>
void	<u>setAudioAttributes</u>
	(https://developer.android.com/reference/android/widget/VideoView.html#setAudioAttributes(android.media.AudioAttribu
	tes))
	( <u>AudioAttributes</u> (https://developer.android.com/reference/android/media/AudioAttributes.html) attributes)
	· · · · · · · · · · · · · · · · · · ·

	Sets the <u>AudioAttributes</u> (https://developer.android.com/reference/android/media/AudioAttributes.html) to be used during the playback of the video.
void	<pre>setAudioFocusRequest (https://developer.android.com/reference/android/widget/VideoView.html#setAudioFocusRequest(int))(int focusGain)</pre>
	Sets which type of audio focus will be requested during the playback, or configures playback to not request audio focus.
void	<pre>setMediaController (https://developer.android.com/reference/android/widget/VideoView.html#setMediaController(android.widget.MediaController)) (MediaController (https://developer.android.com/reference/android/widget/MediaController.html) controller)</pre>
void	<pre>setOnCompletionListener (https://developer.android.com/reference/android/widget/VideoView.html#setOnCompletionListener(android.media.MediaPlayer.OnCompletionListener)) (MediaPlayer.OnCompletionListener (https://developer.android.com/reference/android/media/MediaPlayer.OnCompletionListener.html) 1)</pre>
	Register a callback to be invoked when the end of a media file has been reached during playback.
void	<pre>setOnErrorListener (https://developer.android.com/reference/android/widget/VideoView.html#setOnErrorListener(android.media.MediaPlayer .OnErrorListener)) (MediaPlayer.OnErrorListener (https://developer.android.com/reference/android/media/MediaPlayer.OnErrorListener.html) 1)</pre>
	Register a callback to be invoked when an error occurs during playback or setup.
void	<u>setOnInfoListener</u> (https://developer.android.com/reference/android/widget/VideoView.html#setOnInfoListener(android.media.MediaPlayer. OnInfoListener))

( <u>MediaPlayer.OnInfoListener</u>
-------------------------------------

(https://developer.android.com/reference/android/media/MediaPlayer.OnInfoListener.html) 1)

Register a callback to be invoked when an informational event occurs during playback or setup.

void

### setOnPreparedListener

(https://developer.android.com/reference/android/widget/VideoView.html#setOnPreparedListener(android.media.MediaPlayer.OnPreparedListener))

# (MediaPlayer.OnPreparedListener

(https://developer.android.com/reference/android/media/MediaPlayer.OnPreparedListener.html) 1)

Register a callback to be invoked when the media file is loaded and ready to go.

void

#### setVideoPath

(https://developer.android.com/reference/android/widget/VideoView.html#setVideoPath(java.lang.String))(<u>String</u> (https://developer.android.com/reference/java/lang/String.html) path)

Sets video path.

void

### setVideoURI

(https://developer.android.com/reference/android/widget/VideoView.html#setVideoURI(android.net.Uri,%20java.util.Map<java.lang.String,%20java.lang.String>))

(<u>Uri</u> (https://developer.android.com/reference/android/net/Uri.html) uri, <u>Map</u>

(https://developer.android.com/reference/java/util/Map.html)<a href="mailto:string">String</a>

(https://developer.android.com/reference/java/lang/String.html), String

(https://developer.android.com/reference/java/lang/String.html)> headers)

Sets video URI using specific headers.

void

<u>setVideoURI</u> (https://developer.android.com/reference/android/widget/VideoView.html#setVideoURI(android.net.Uri))
(<u>Uri</u> (https://developer.android.com/reference/android/net/Uri.html) uri)

Sets video URI.

void	<u>start</u> (https://developer.android.com/reference/android/widget/VideoView.html#start())()
void	<pre>stopPlayback (https://developer.android.com/reference/android/widget/VideoView.html#stopPlayback())()</pre>
void	<pre>suspend (https://developer.android.com/reference/android/widget/VideoView.html#suspend())()</pre>

### Protected methods

voidonAttachedToWindow (https://developer.android.com/reference/android/widget/VideoView.html#onAttachedToWindow())()

This is called when the view is attached to a window.

voidonDetachedFromWindow (https://developer.android.com/reference/android/widget/VideoView.html#onDetachedFromWindow())()

This is called when the view is detached from a window.

voidonLayout (https://developer.android.com/reference/android/widget/VideoView.html#onLayout(boolean,%20int,%20int,%20int,%20int))(boolean
changed, int left, int top, int right, int bottom)

Called from layout when this view should assign a size and position to each of its children.

voidonMeasure (https://developer.android.com/reference/android/widget/VideoView.html#onMeasure(int,%20int))(int widthMeasureSpec, int heightMeasureSpec)

Measure the view and its content to determine the measured width and the measured height.

# Inherited methods

From class <a href="mailto:android.view.SurfaceView">android.view.SurfaceView</a> (https://developer.android.com/reference/android/view/SurfaceView.html)

void <u>dispatchDraw</u>

void Ca	alled by draw to draw the child views.
void	
	<u>aw</u> ttps://developer.android.com/reference/android/view/
( <u>Ca</u>	s.Canvas))  anvas (https://developer.android.com/reference/andr
	nvas) Inually render this view (and all of its children) to the g
boolean gat	<u>therTransparentRegion</u>
	ttps://developer.android.com/reference/android/view/
Reg	gion(android.graphics.Region))
( <u>Re</u>	egion (https://developer.android.com/reference/andr
This	is is used by the RootView to perform an optimization
or s	several SurfaceView.
<u>SurfaceHolder</u> (https://developer.android.com/reference/android/view/SurfaceHolder.html) <u>get</u>	<u>tHolder</u>
(htt	ttps://developer.android.com/reference/android/view/
Retu	turn the SurfaceHolder providing access and control o
surf	face.
void	<u>AttachedToWindow</u>
(htt dow	ttps://developer.android.com/reference/android/view/
()	···(/)

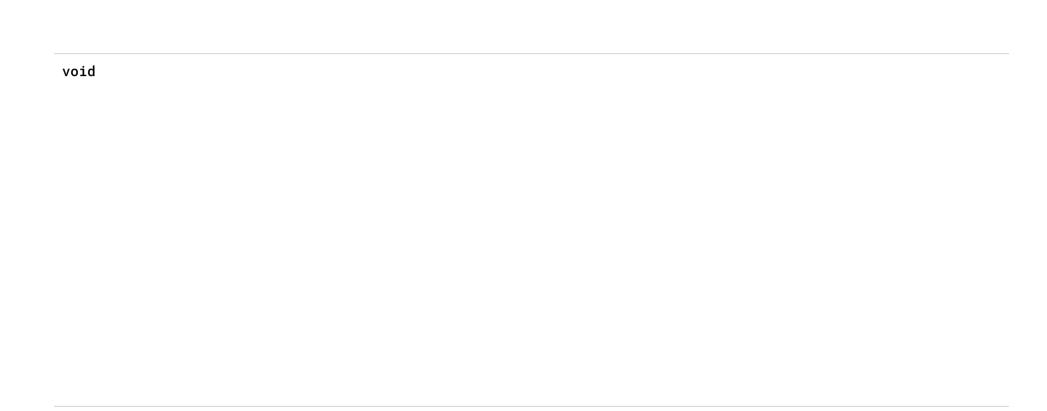
	This is called when the view is attached to a window.
void	<pre>onDetachedFromWindow   (https://developer.android.com/reference/android/view/ Window()) ( )</pre>
	This is called when the view is detached from a window.
void	<pre>onMeasure   (https://developer.android.com/reference/android/view/ Oint))   (int widthMeasureSpec, int heightMeasureSp</pre>
	Measure the view and its content to determine the measu
void	<pre>onWindowVisibilityChanged   (https://developer.android.com/reference/android/view/ yChanged(int))   (int visibility)</pre>
	Called when the window containing has change its visibil (https://developer.android.com/reference/android/view/ (https://developer.android.com/reference/android/view/ (https://developer.android.com/reference/android/view/
void	<pre>setSecure (https://developer.android.com/reference/android/view/ n)) (boolean isSecure)</pre>
	Control whether the surface view's content should be trea appearing in screenshots or from being viewed on non-sa

void	<pre>setVisibility (https://developer.android.com/reference/android/view/ (int visibility)</pre>
	Set the visibility state of this view.
void	<u>setZOrderMediaOverlay</u>
	(https://developer.android.com/reference/android/view/
	verlay(boolean))
	(boolean isMediaOverlay)
	Control whether the surface view's surface is placed on to
	the window (but still behind the window itself).
void	<u>setZOrderOnTop</u>
void	<u>setZ0rder0nTop</u> (https://developer.android.com/reference/android/view
void	
void	(https://developer.android.com/reference/android/view

From class <a href="mailto:android.view.View">android.view.View</a>. (https://developer.android.com/reference/android/view/View.html)

void

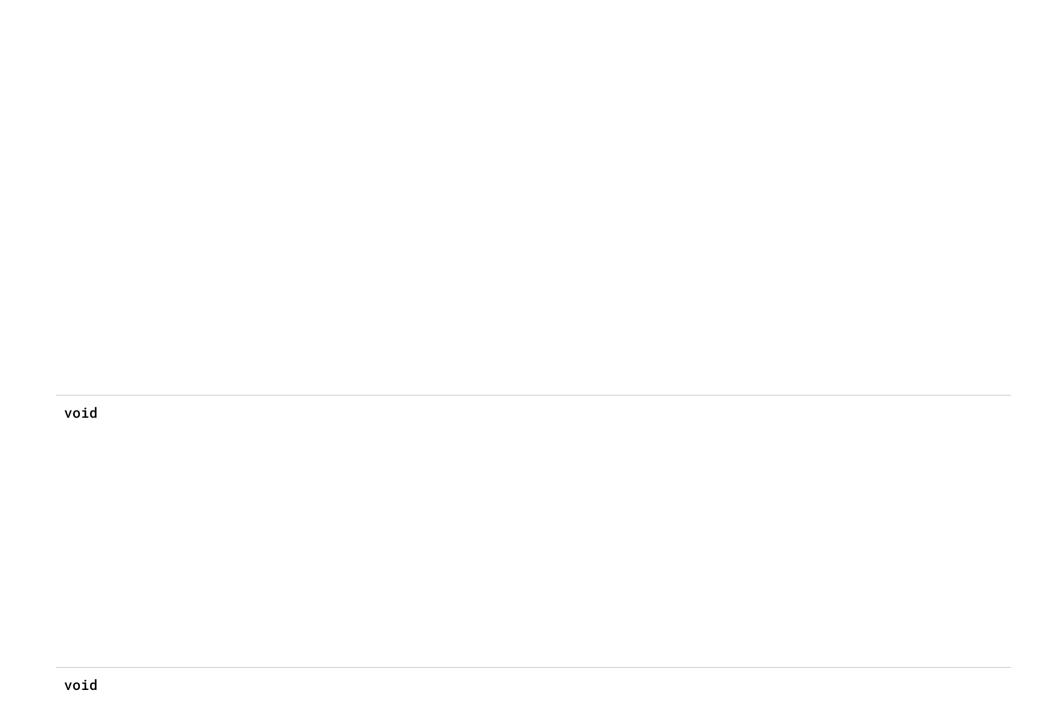


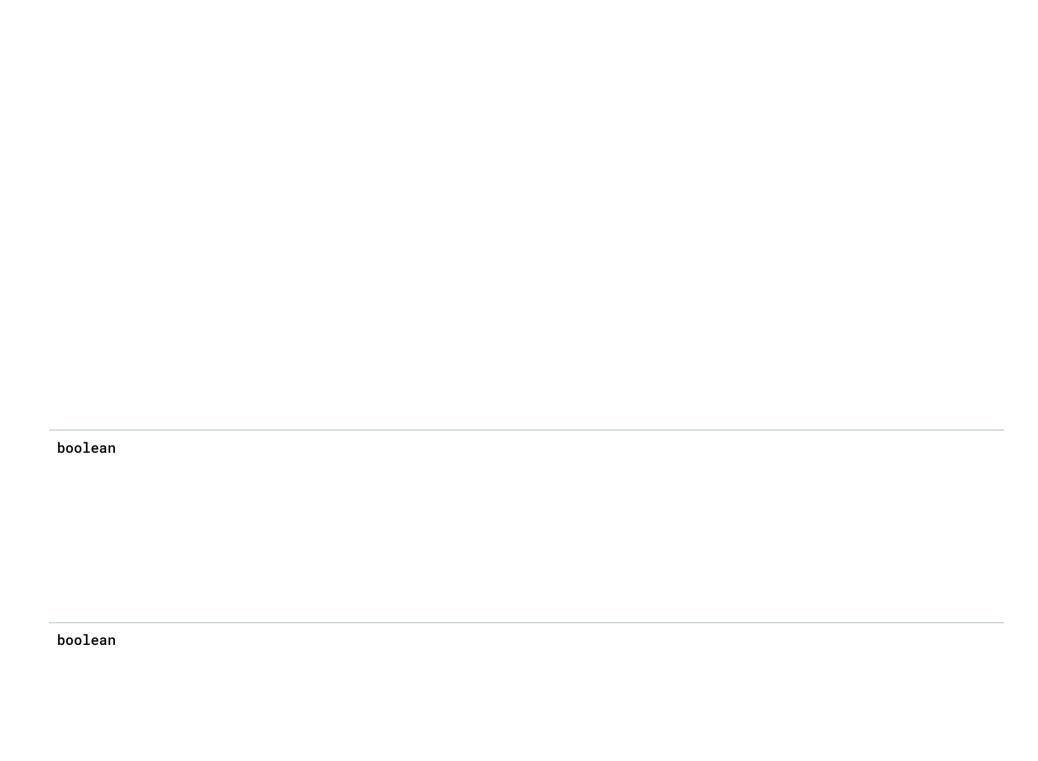


void

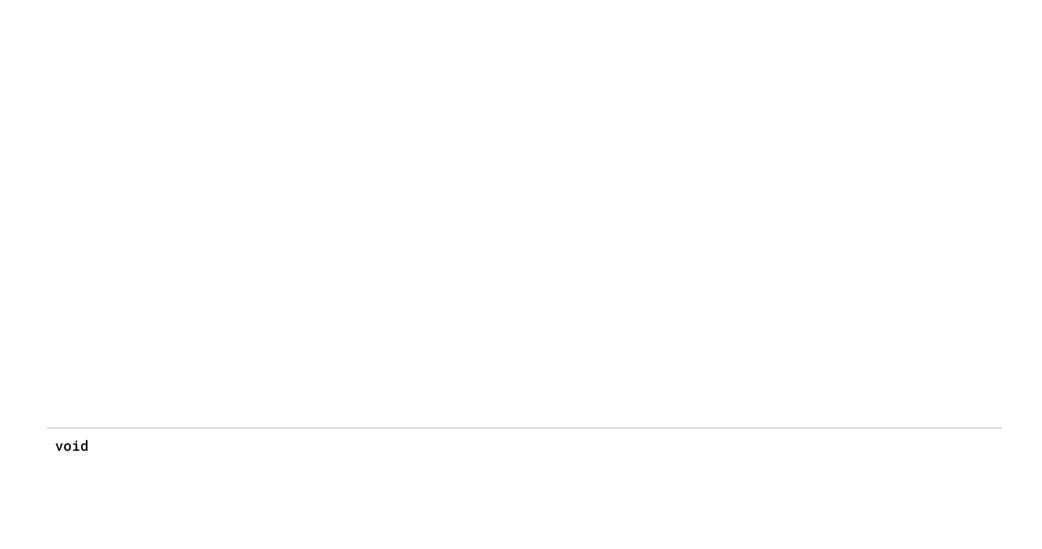


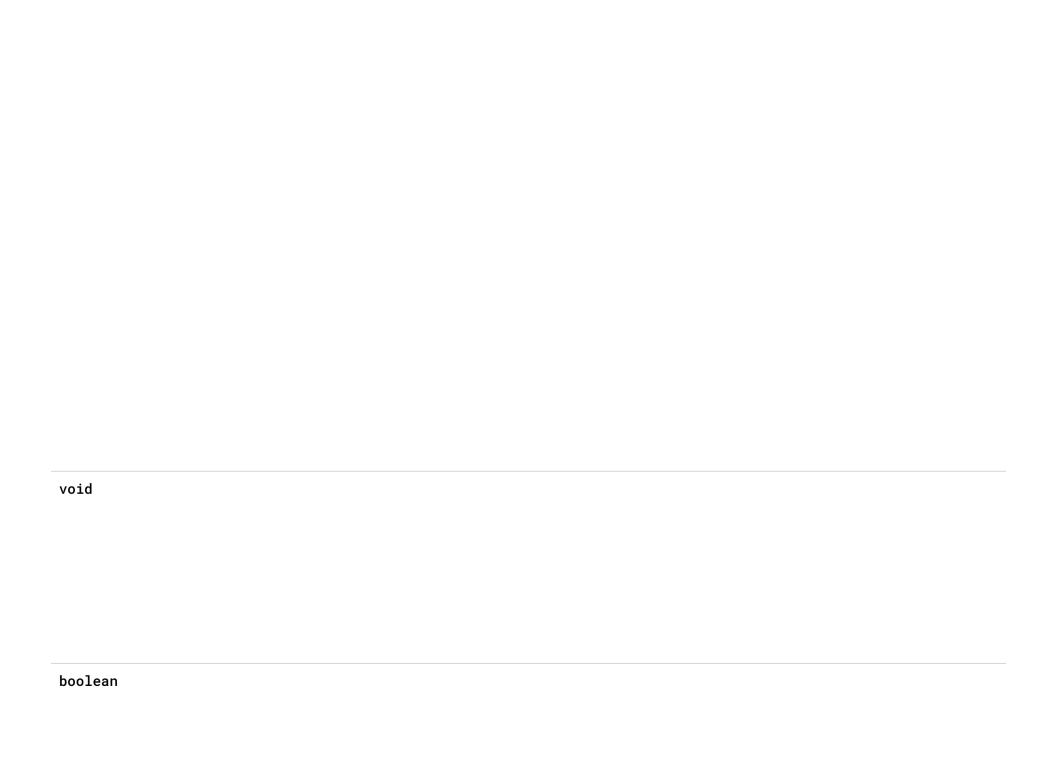


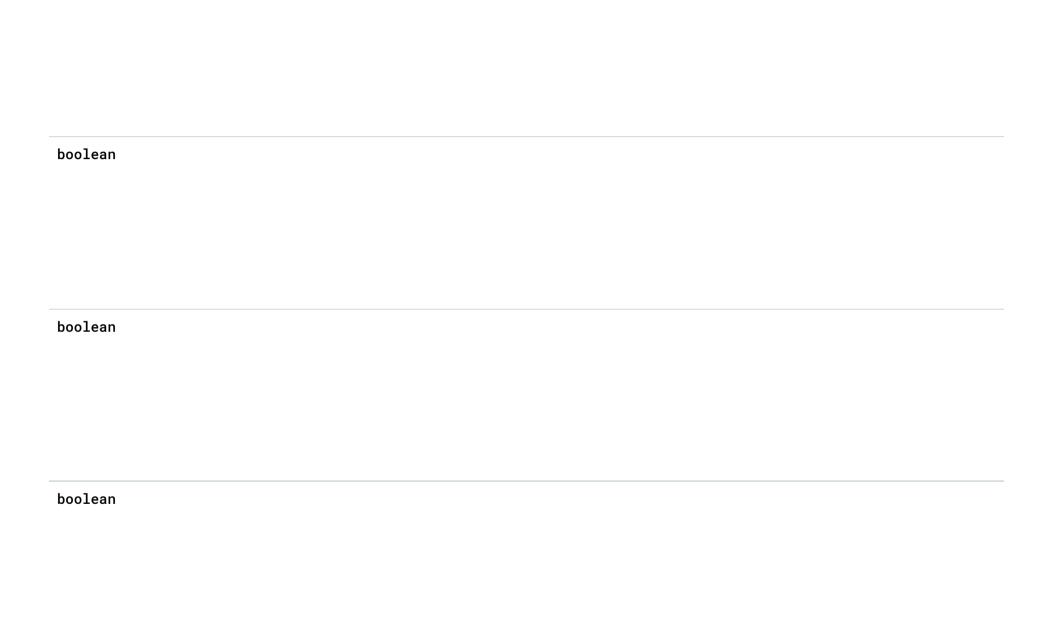


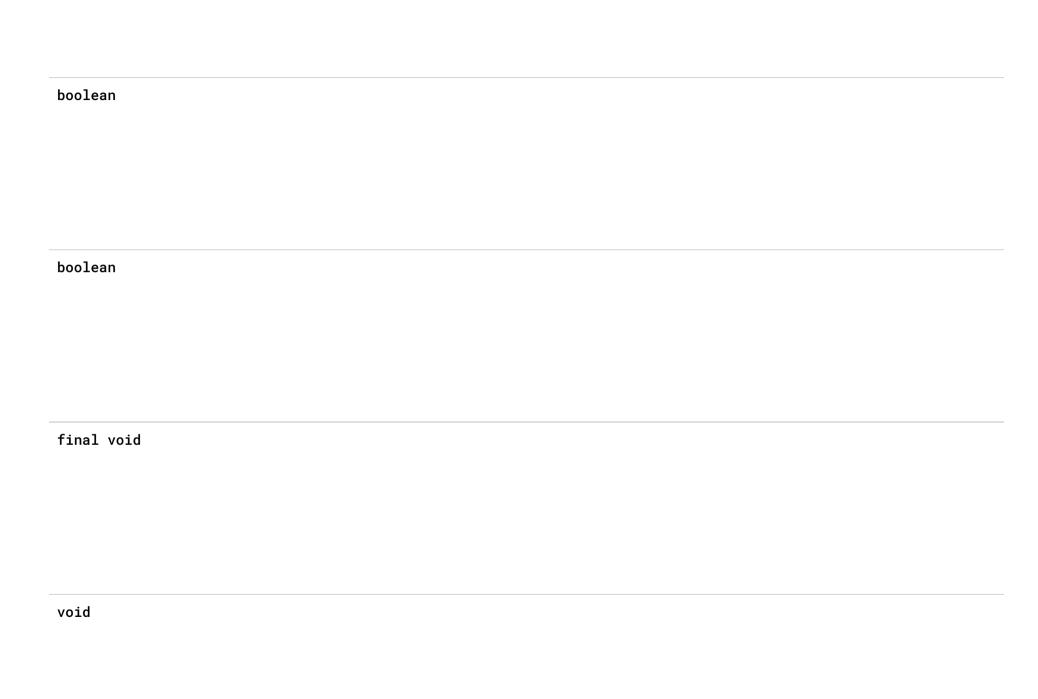


boolean		
void		
void		



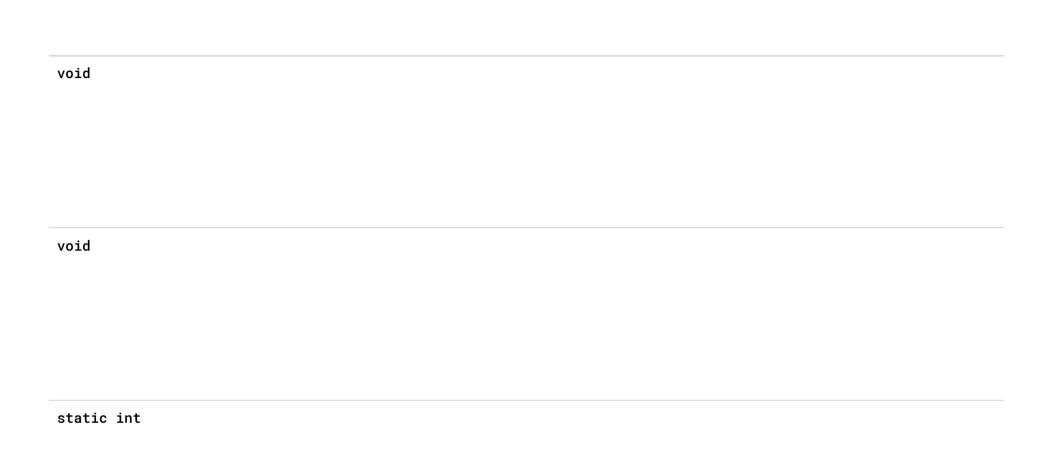


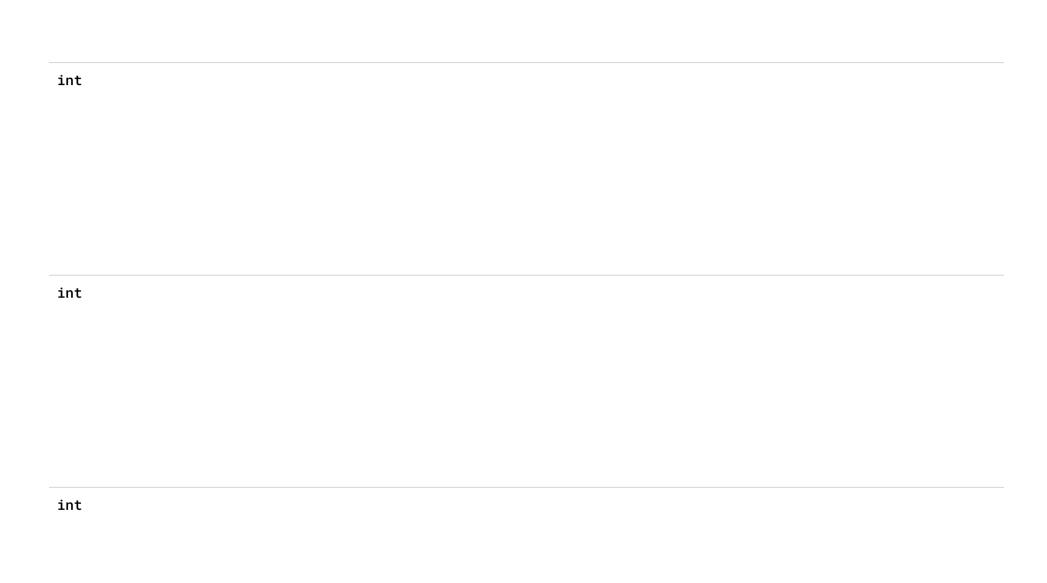


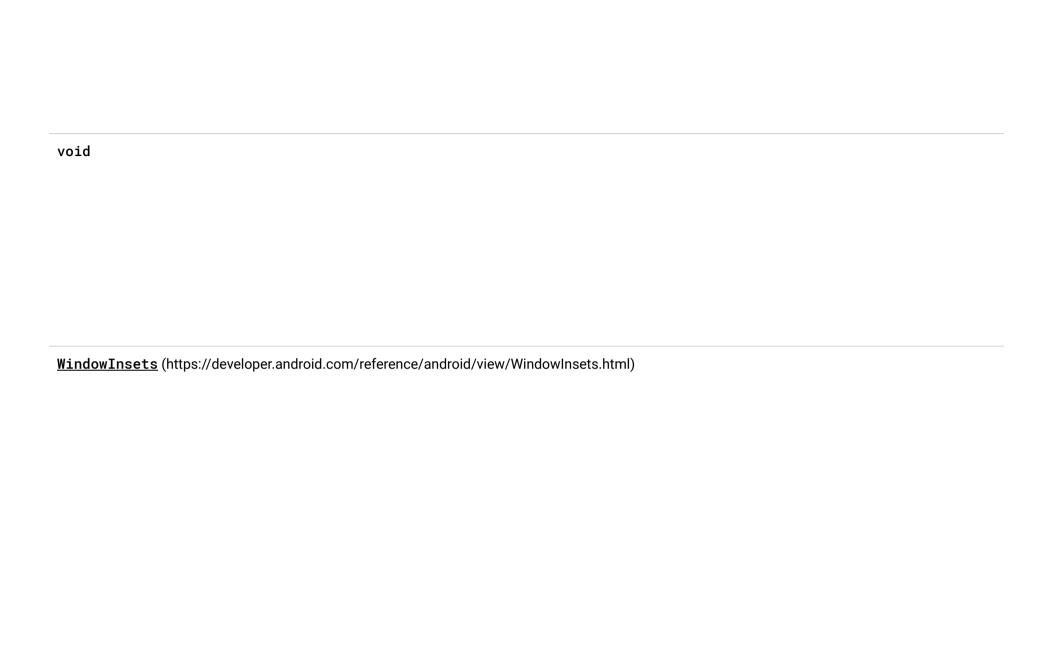


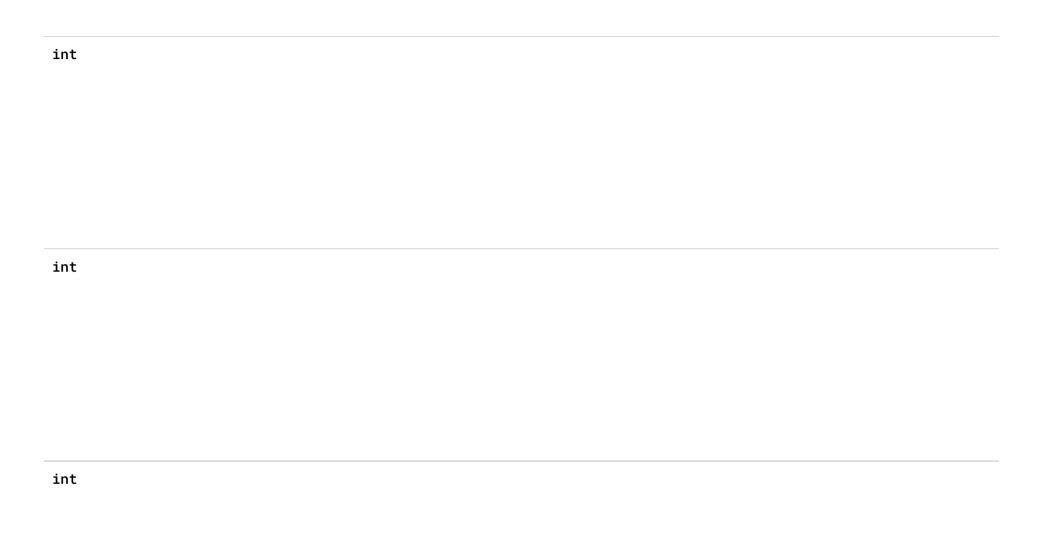
final void

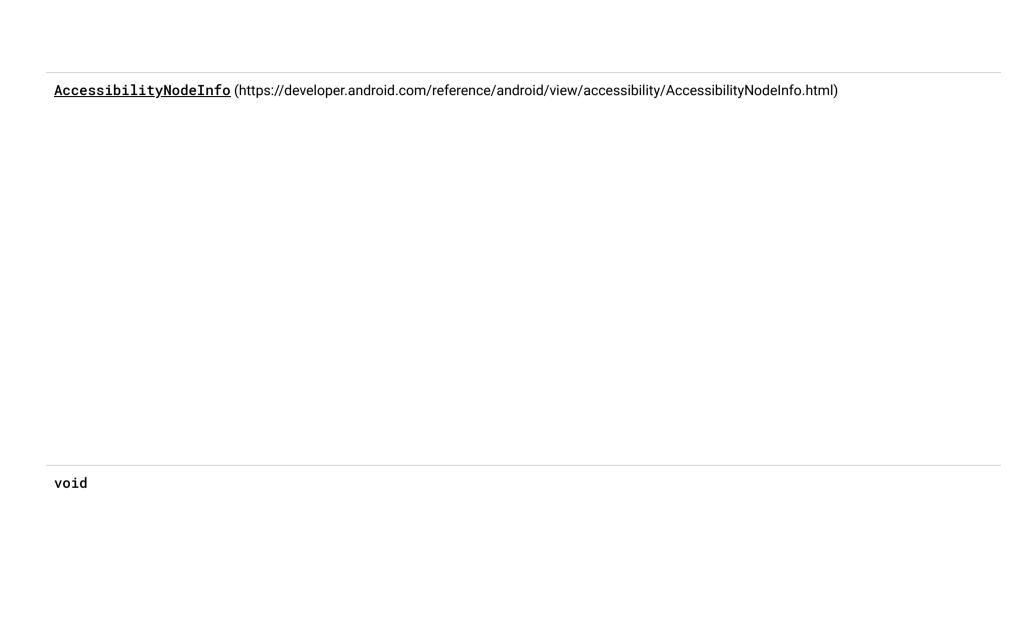
boolean

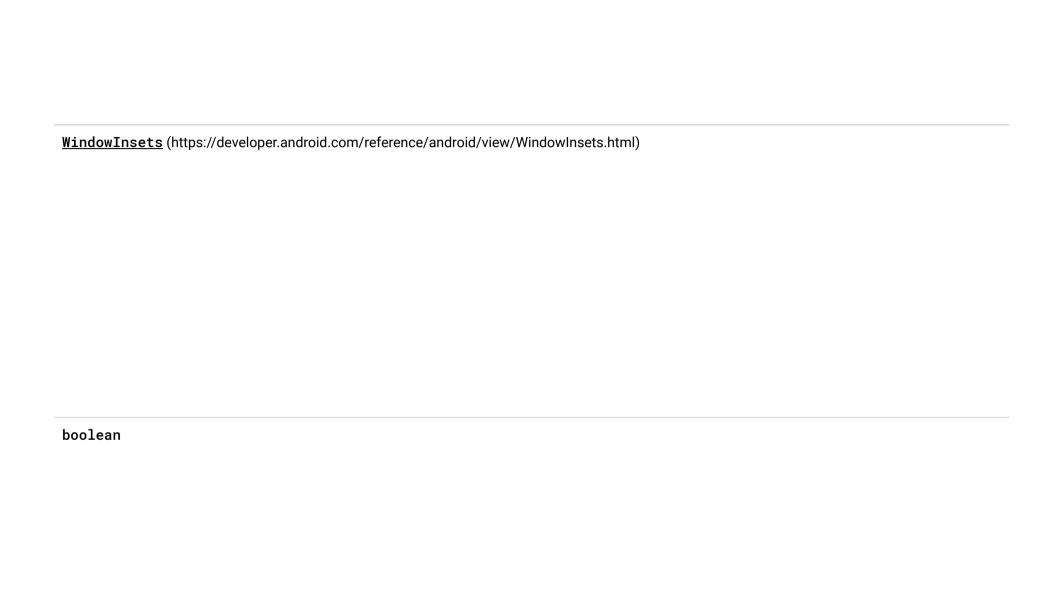


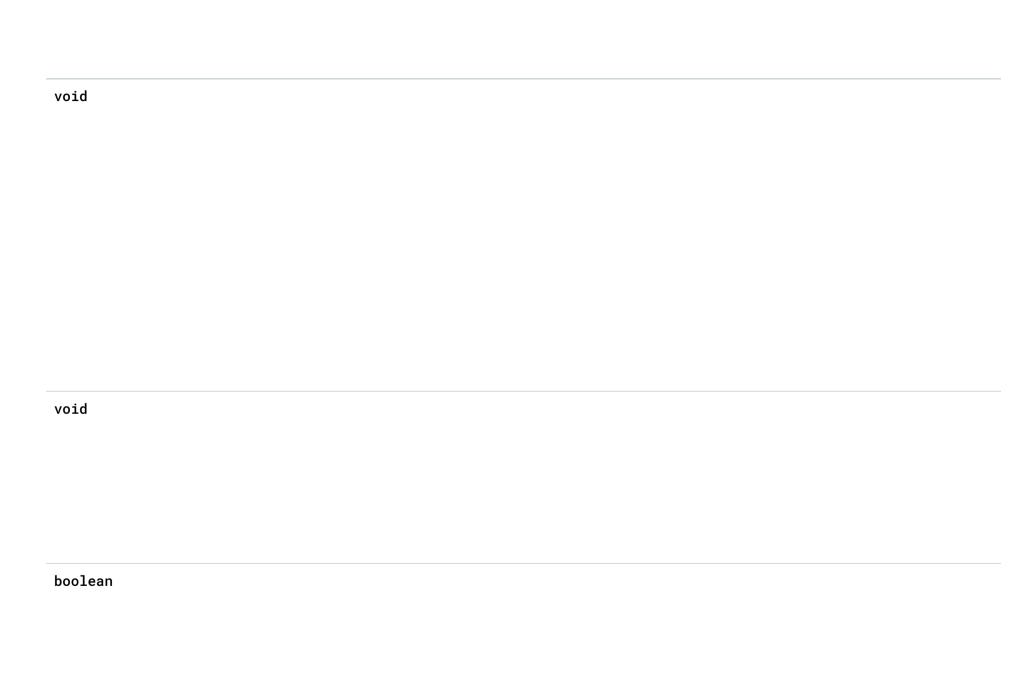


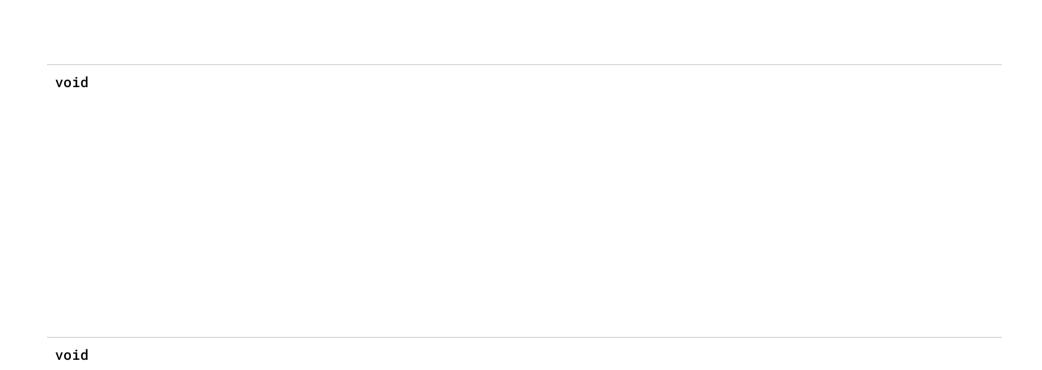


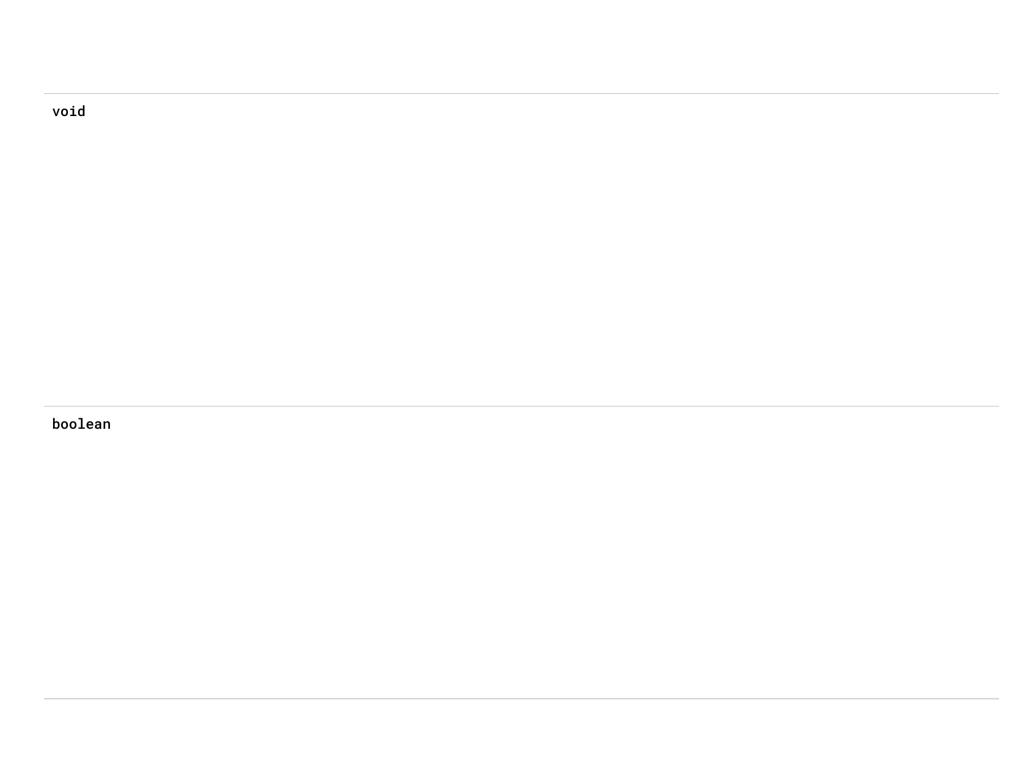


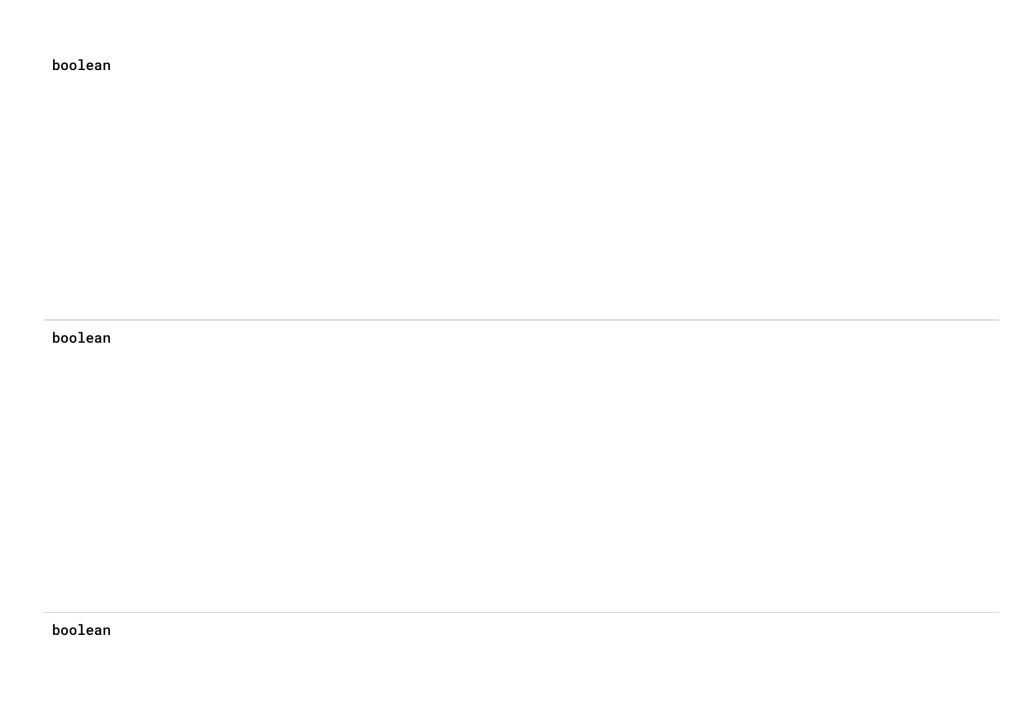


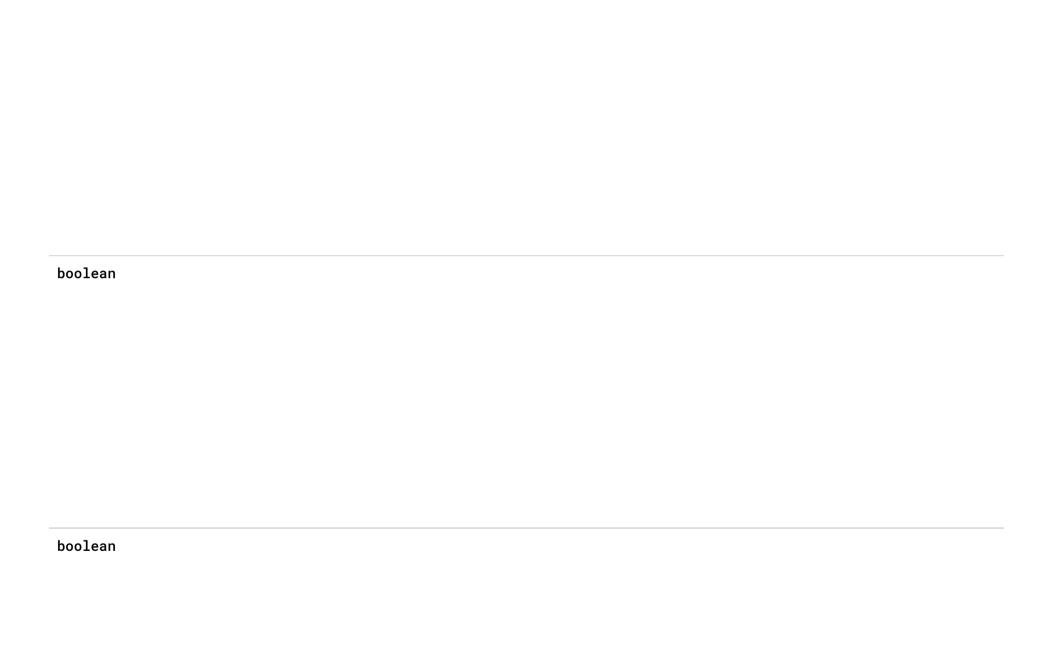


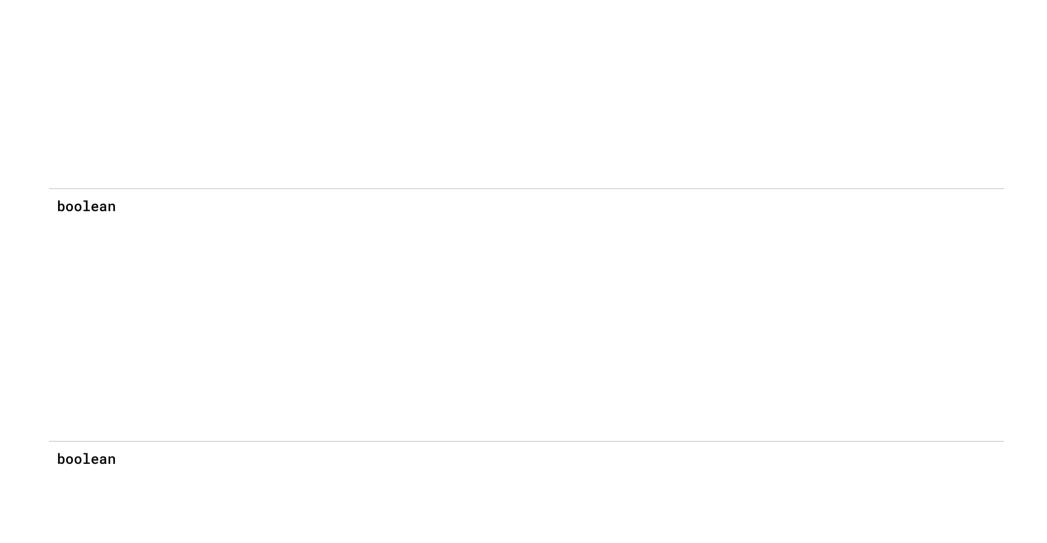


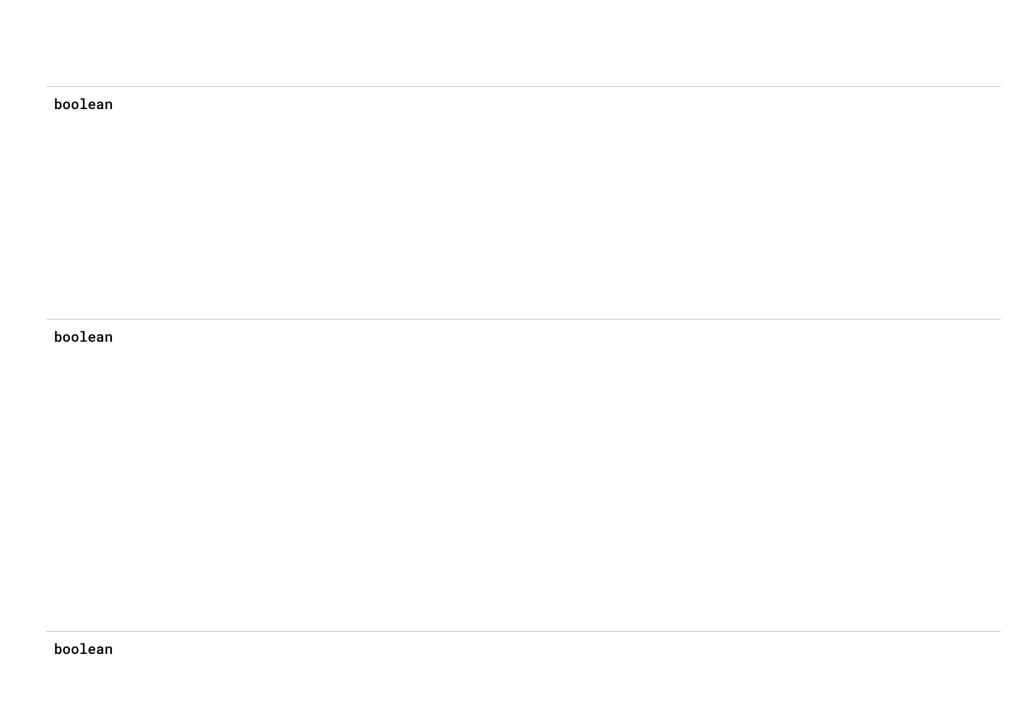


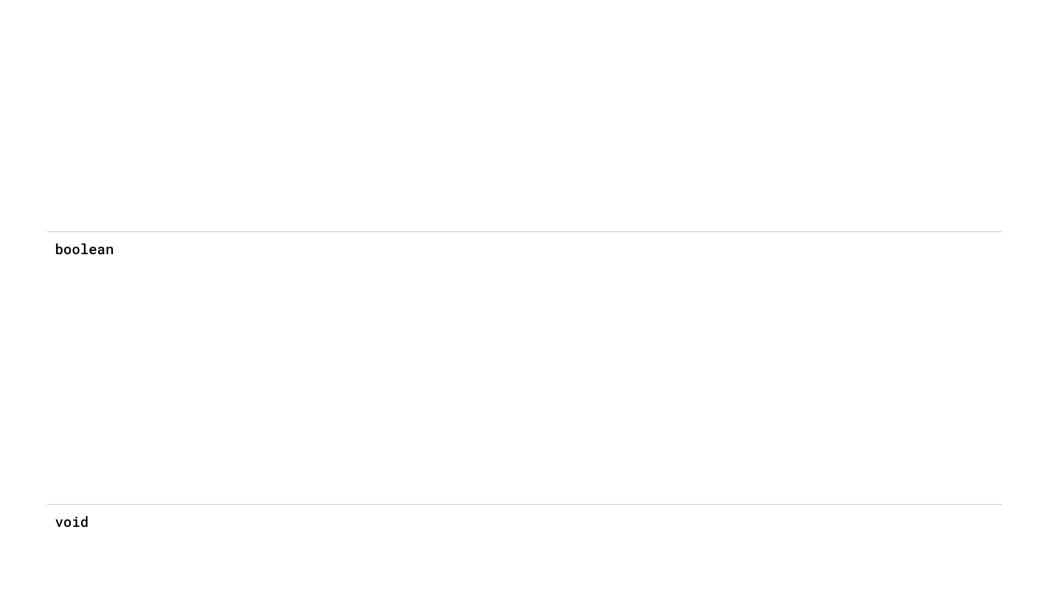






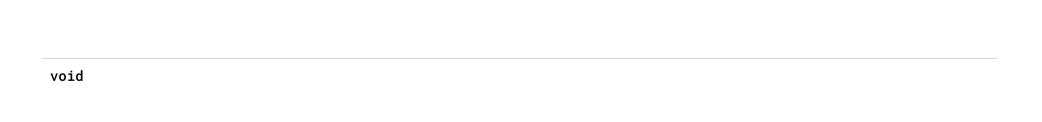


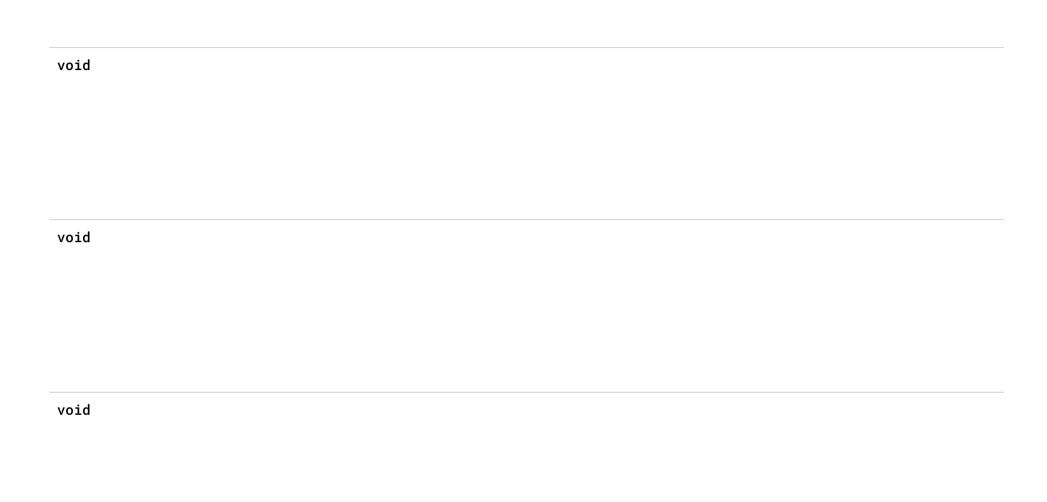




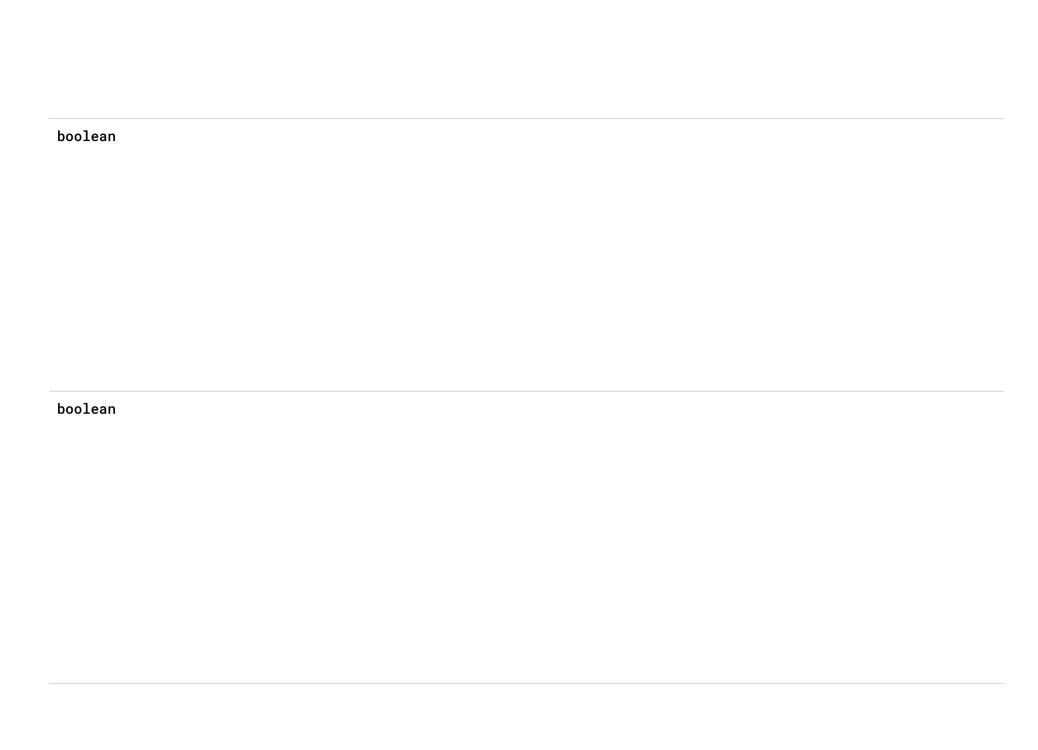
boolean

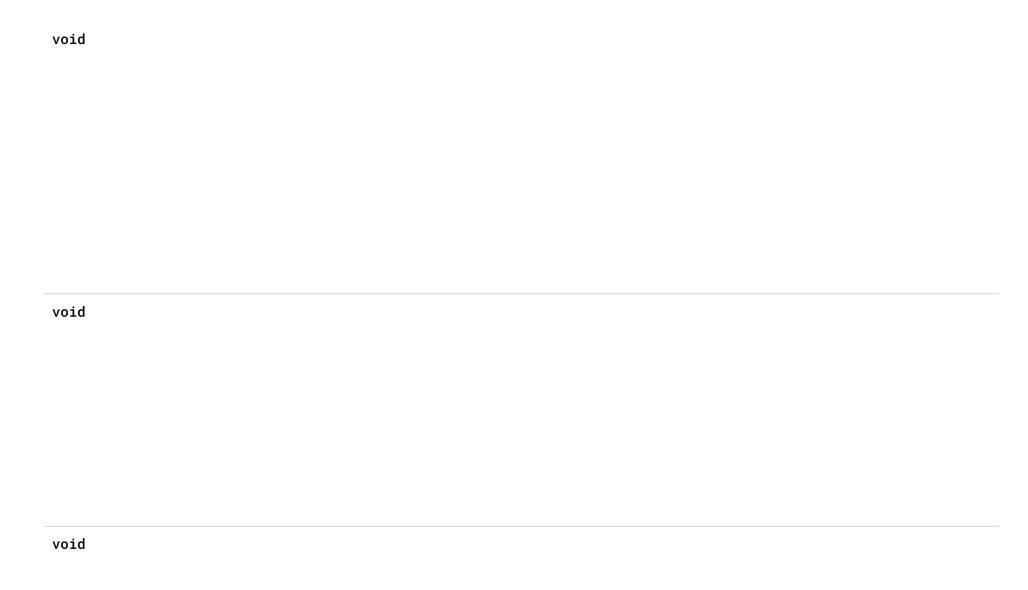


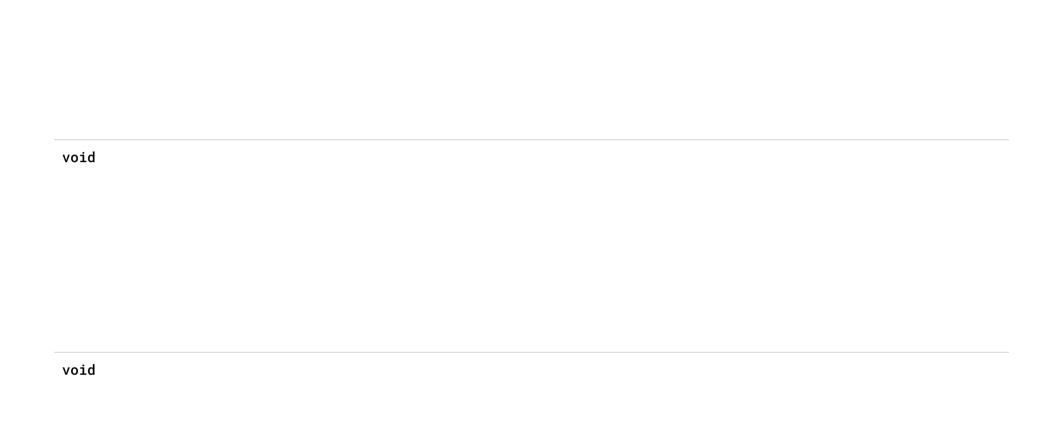


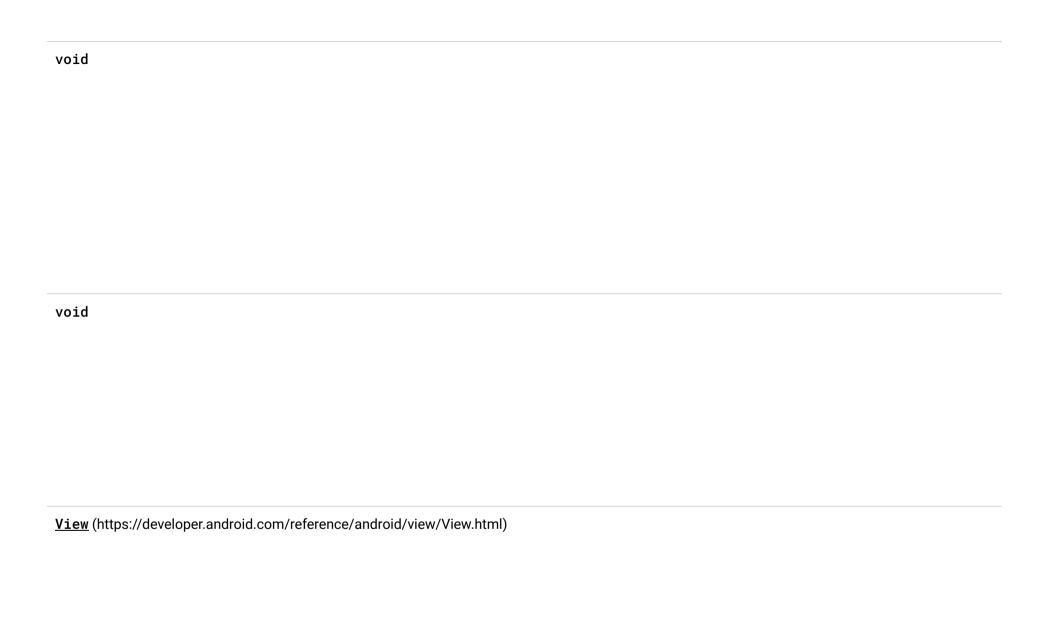


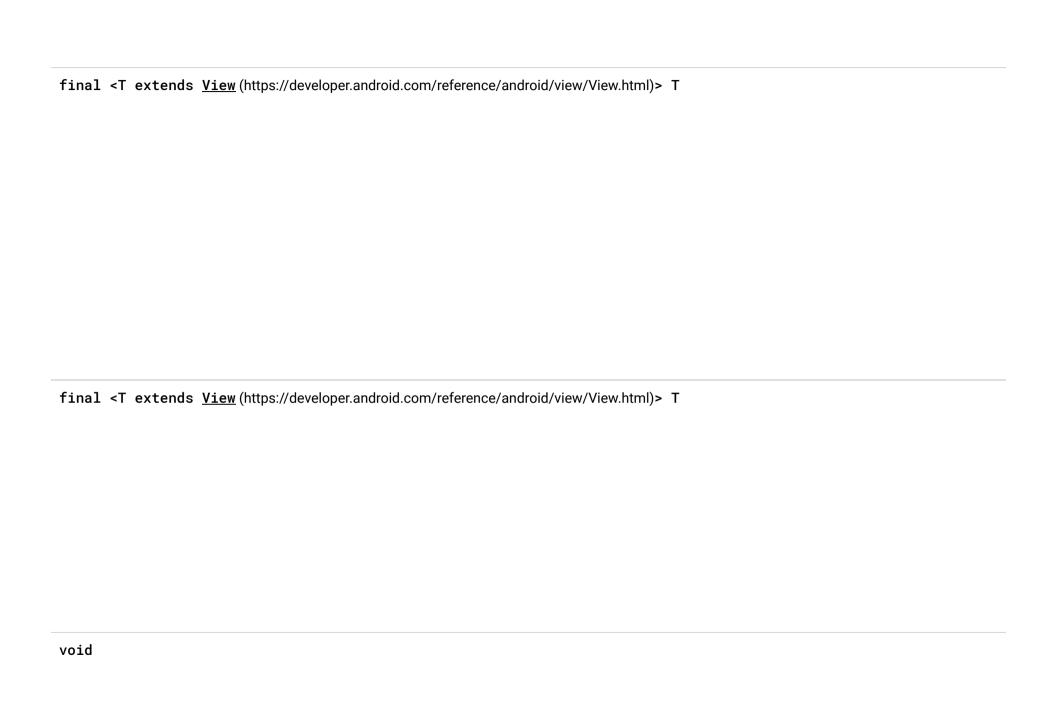
boolean

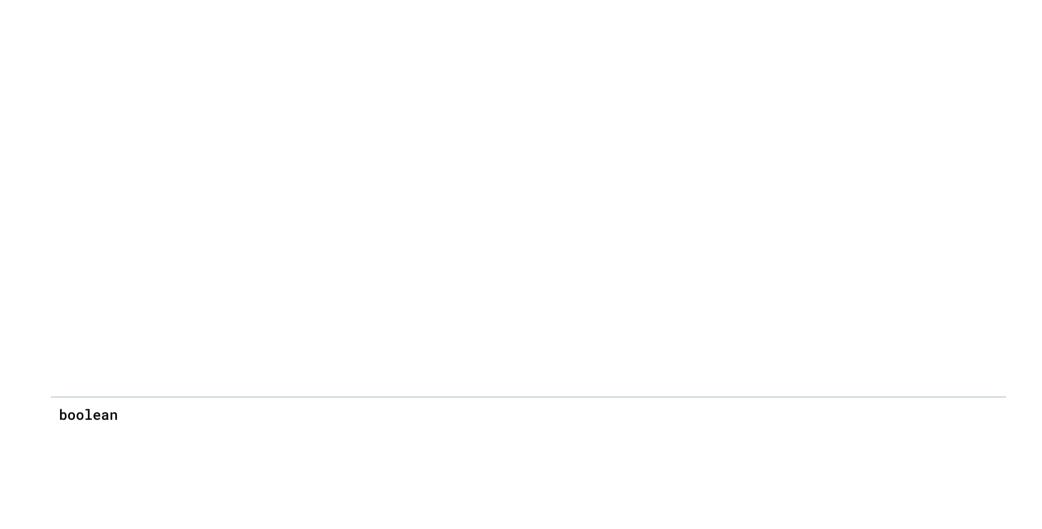




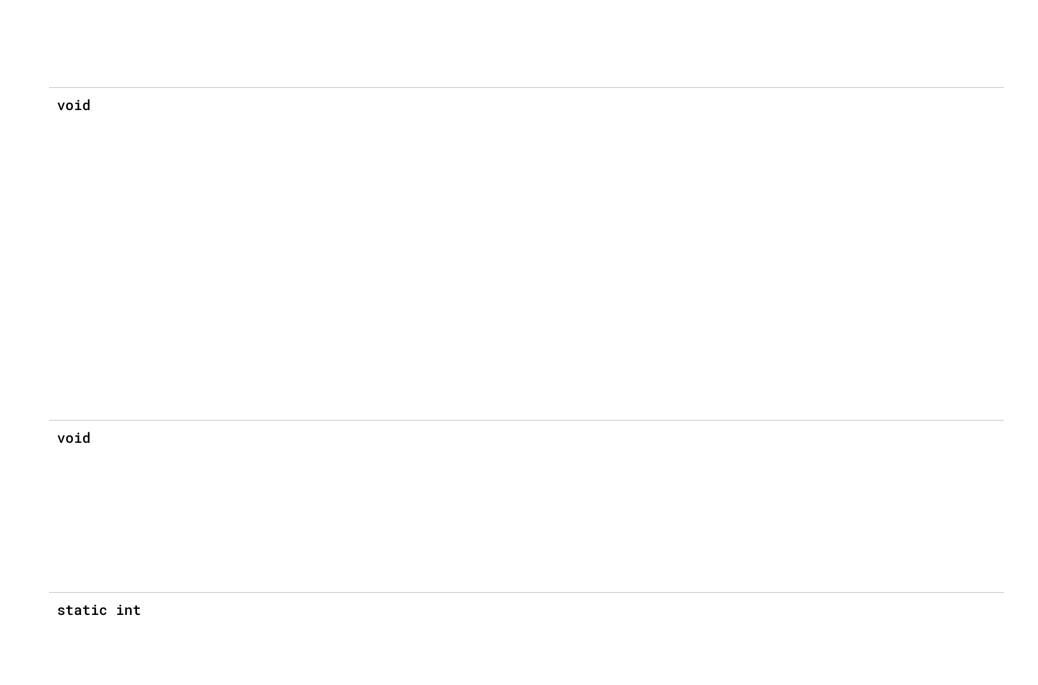


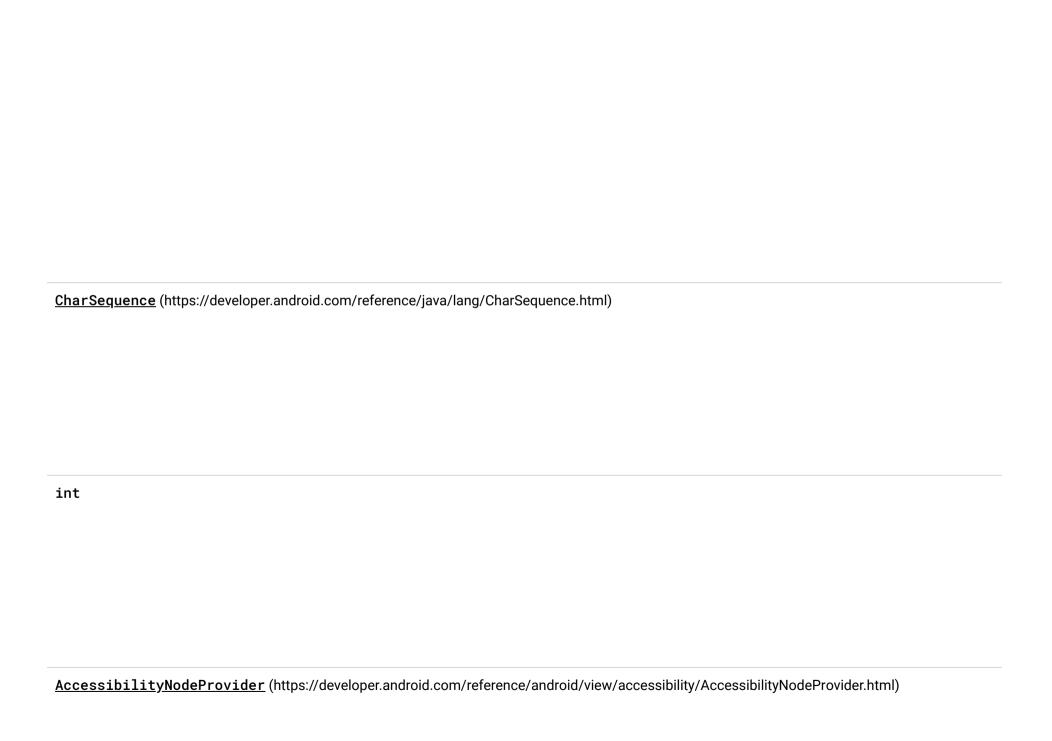




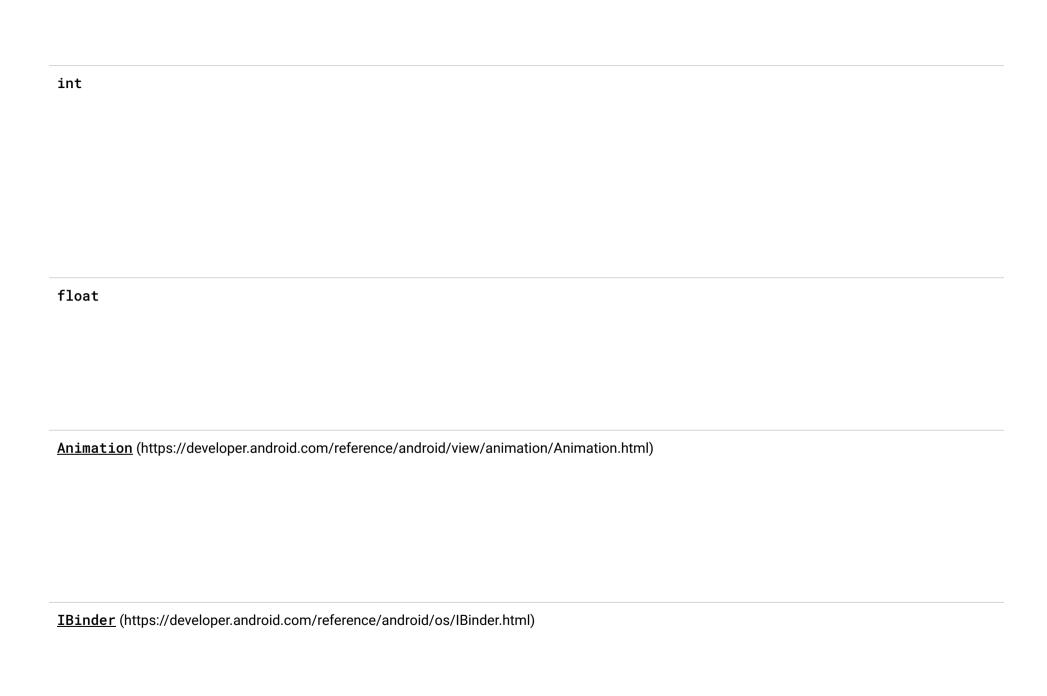


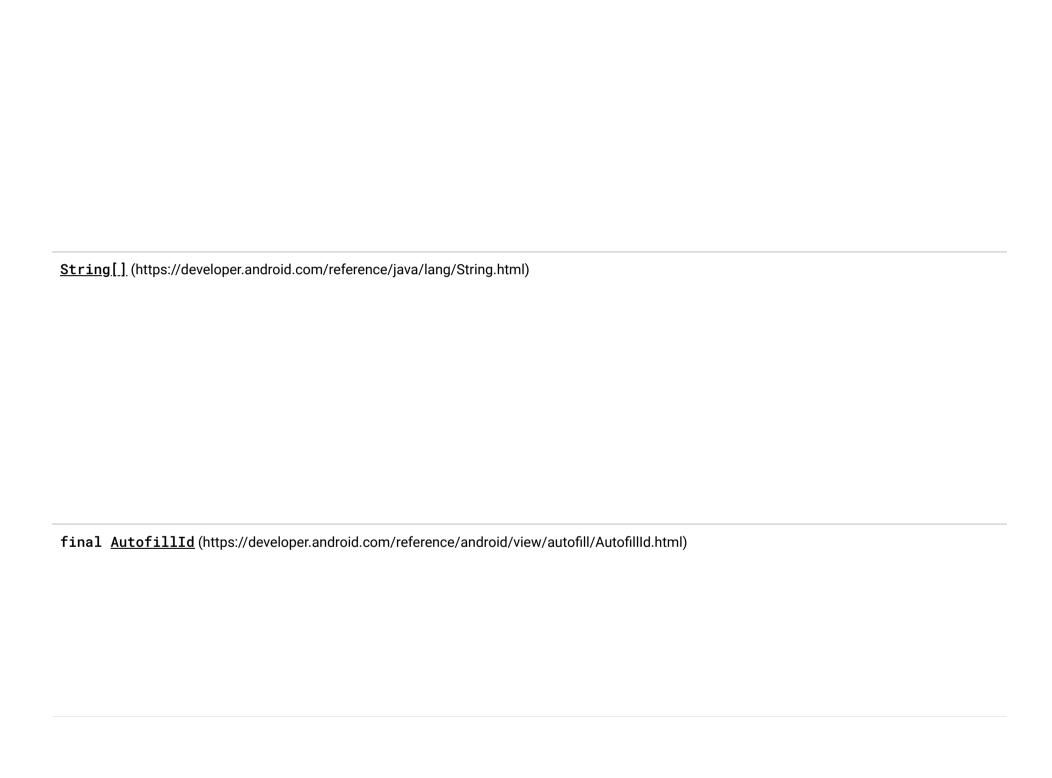


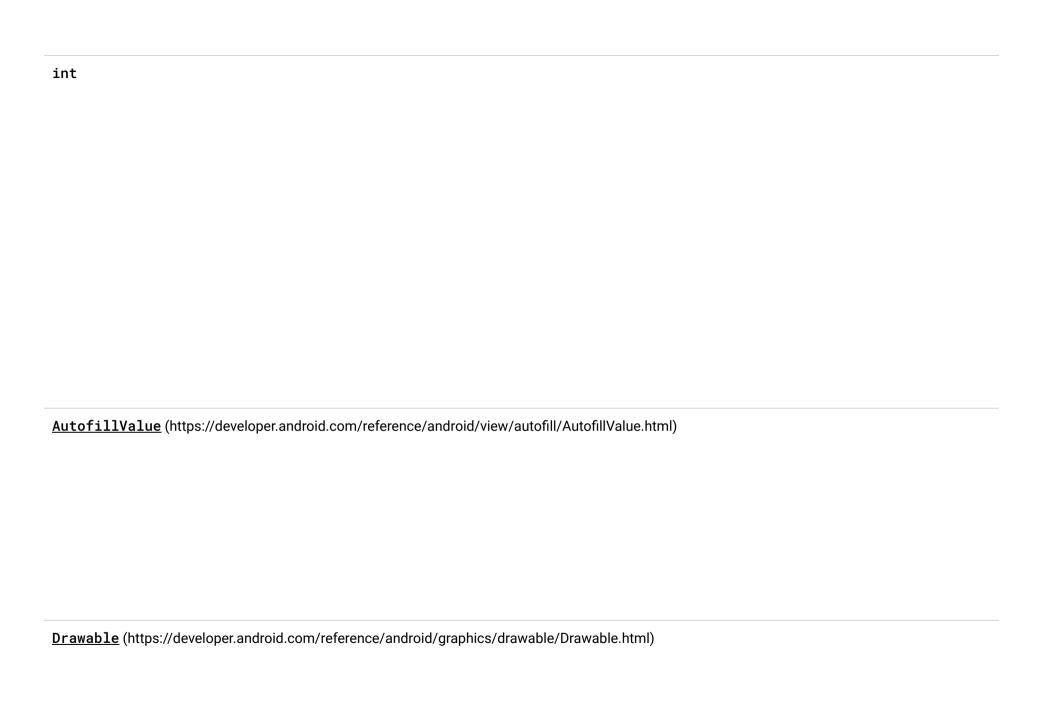




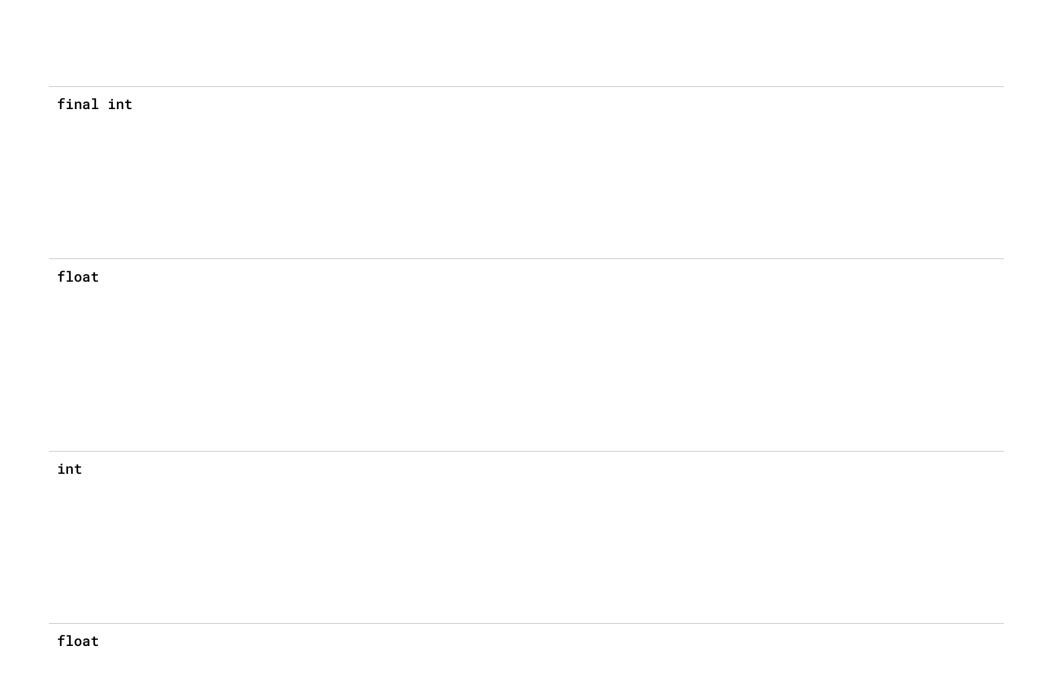




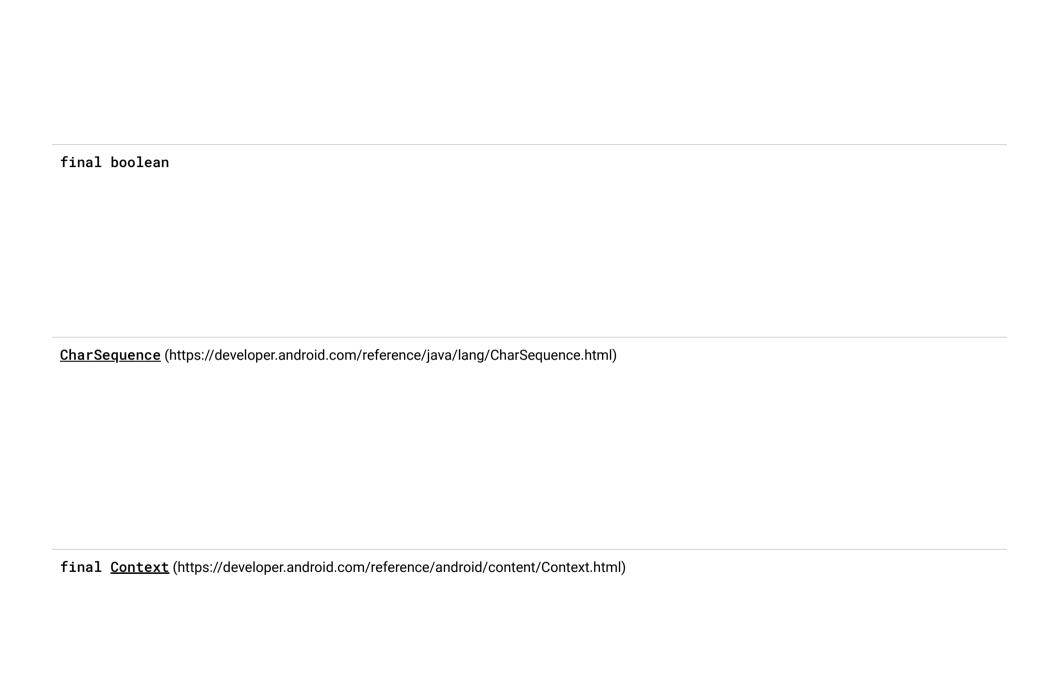


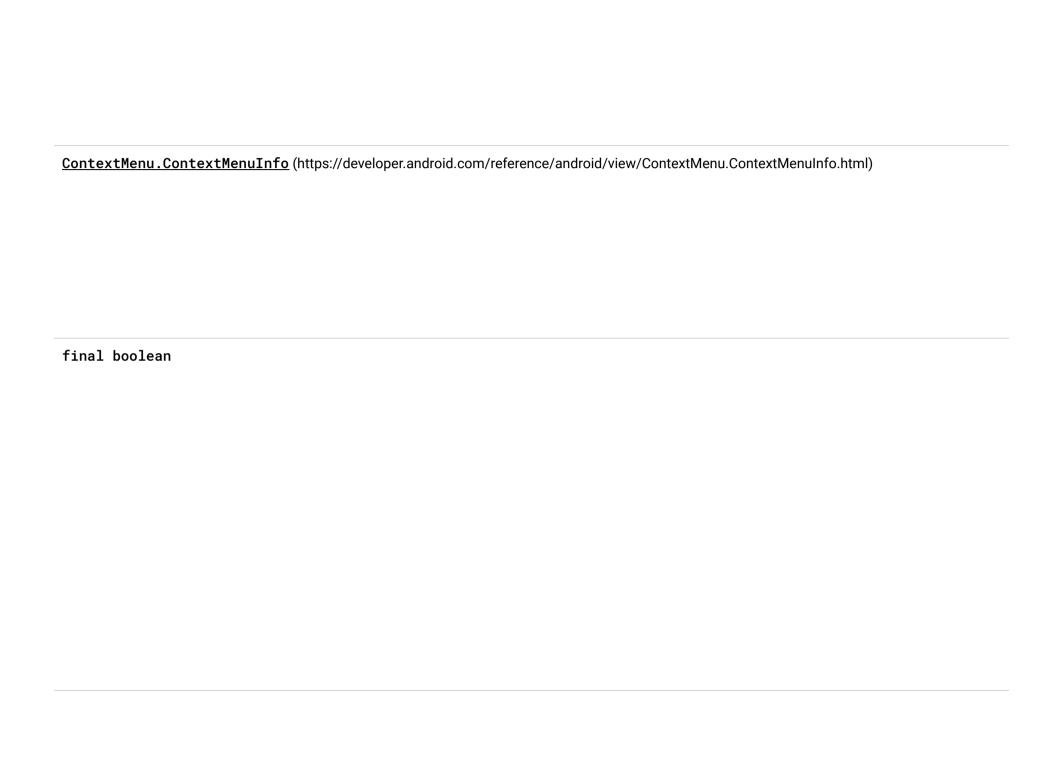


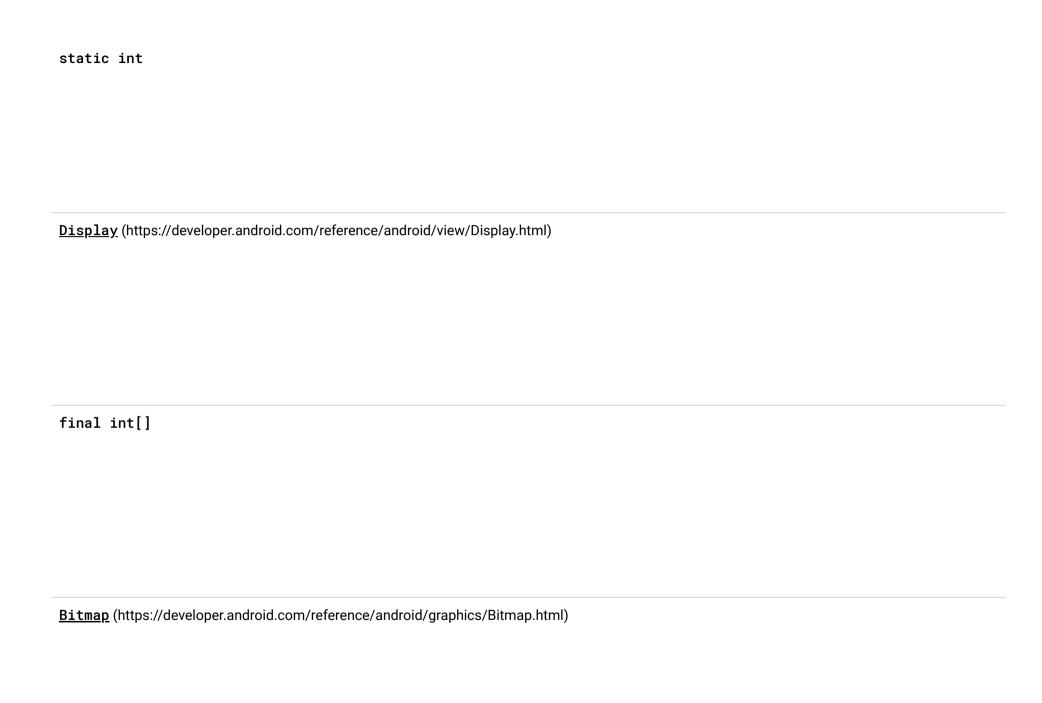
<u>ColorStateList</u> (https://developer.android.com/reference/android/content/res/ColorStateList.html)
<u>PorterDuff.Mode</u> (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)
<u>PorterDuff.Mode</u> (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)
<u>PorterDuff.Mode</u> (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)
PorterDuff.Mode (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)
PorterDuff.Mode (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)
PorterDuff.Mode (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)
PorterDuff.Mode (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)  int







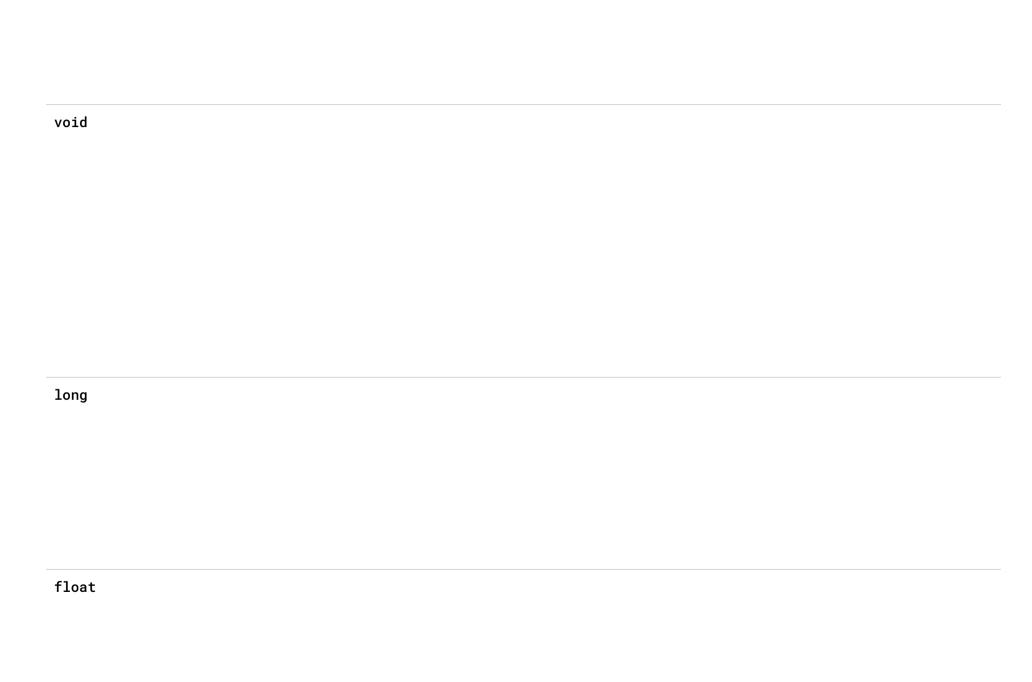


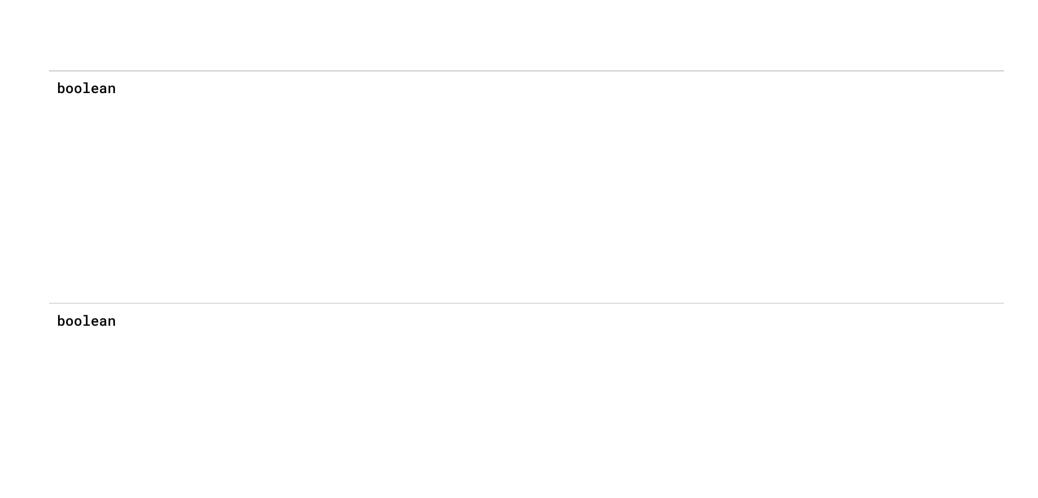


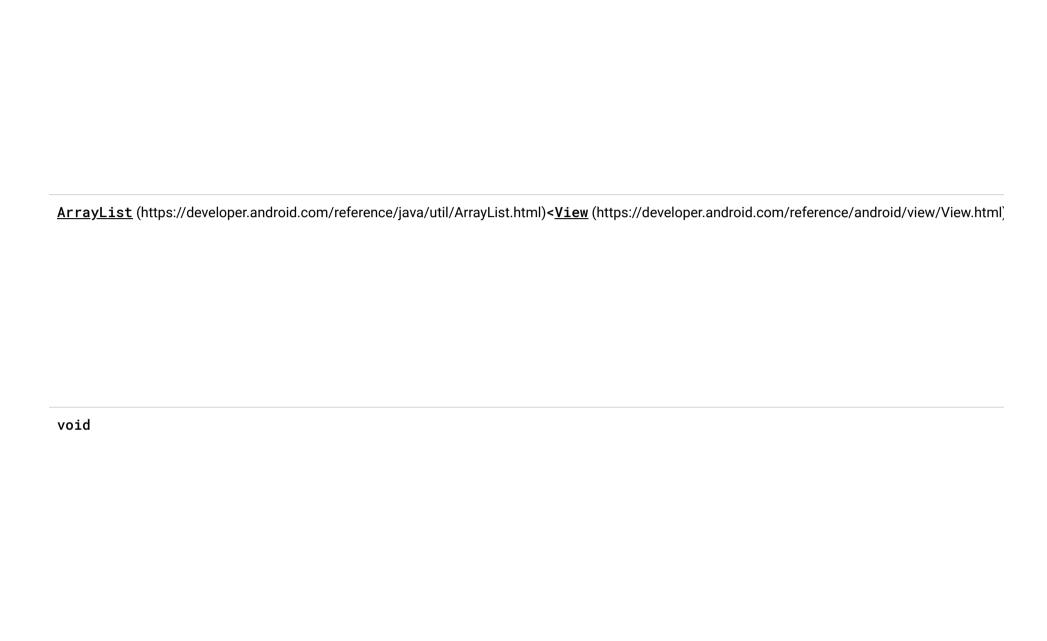


int

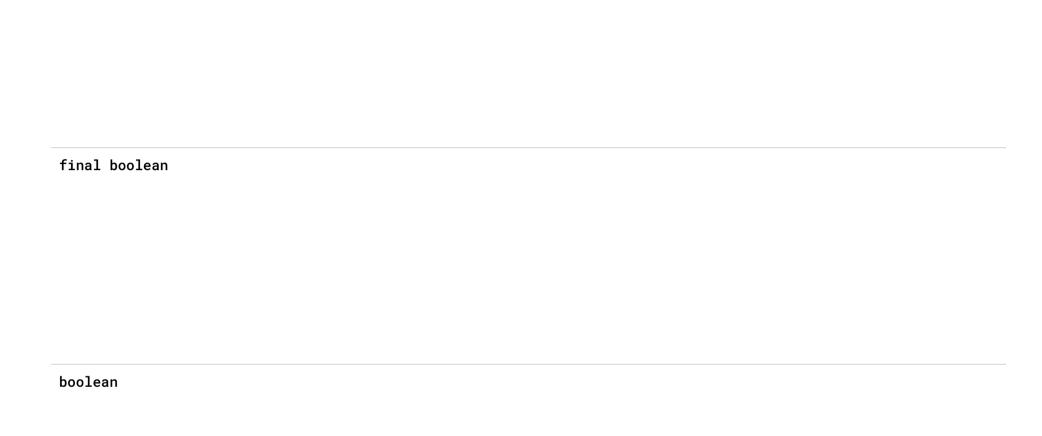
int



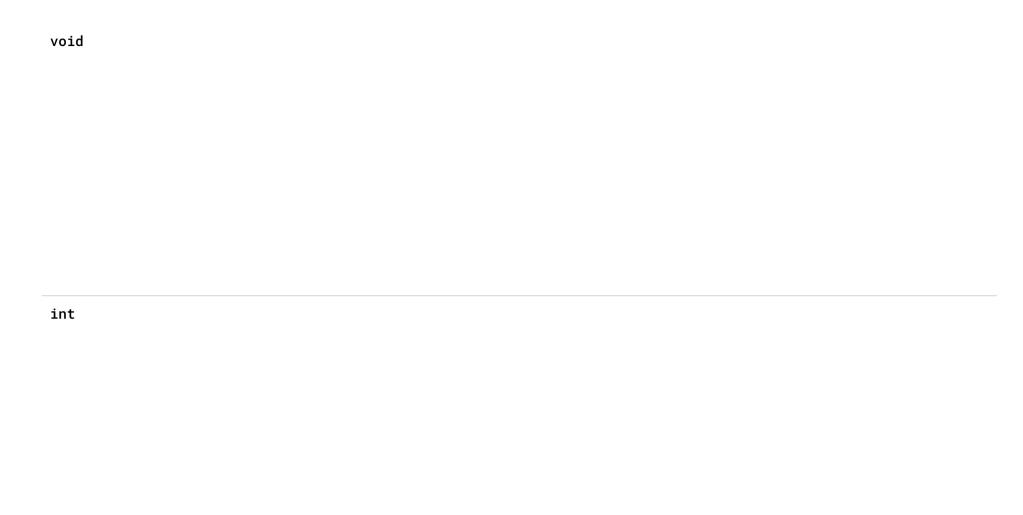


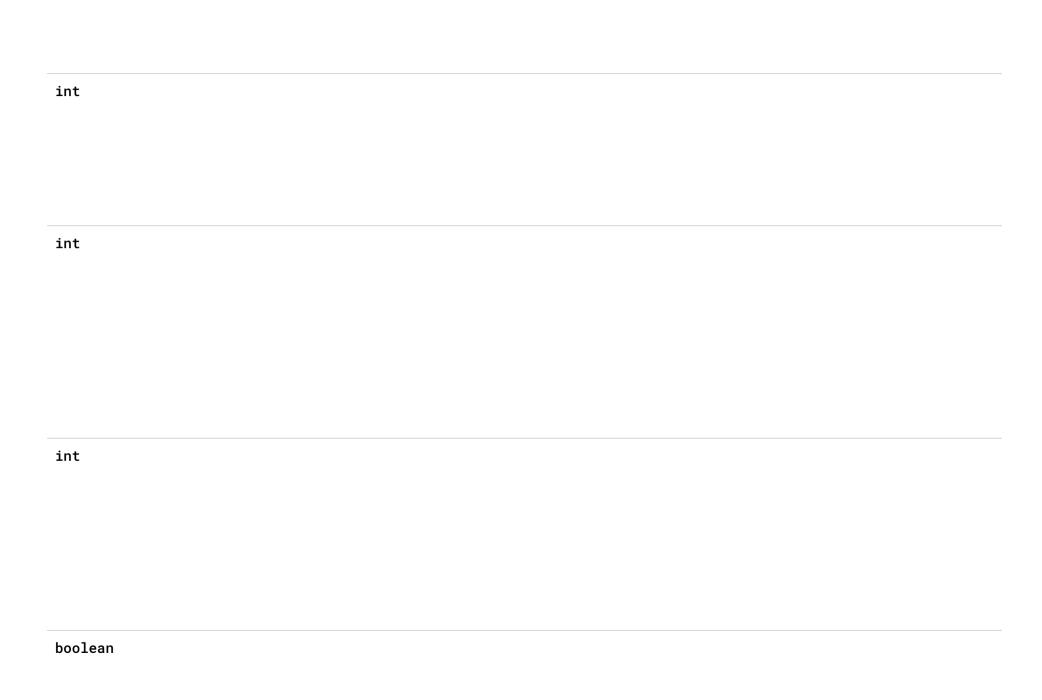


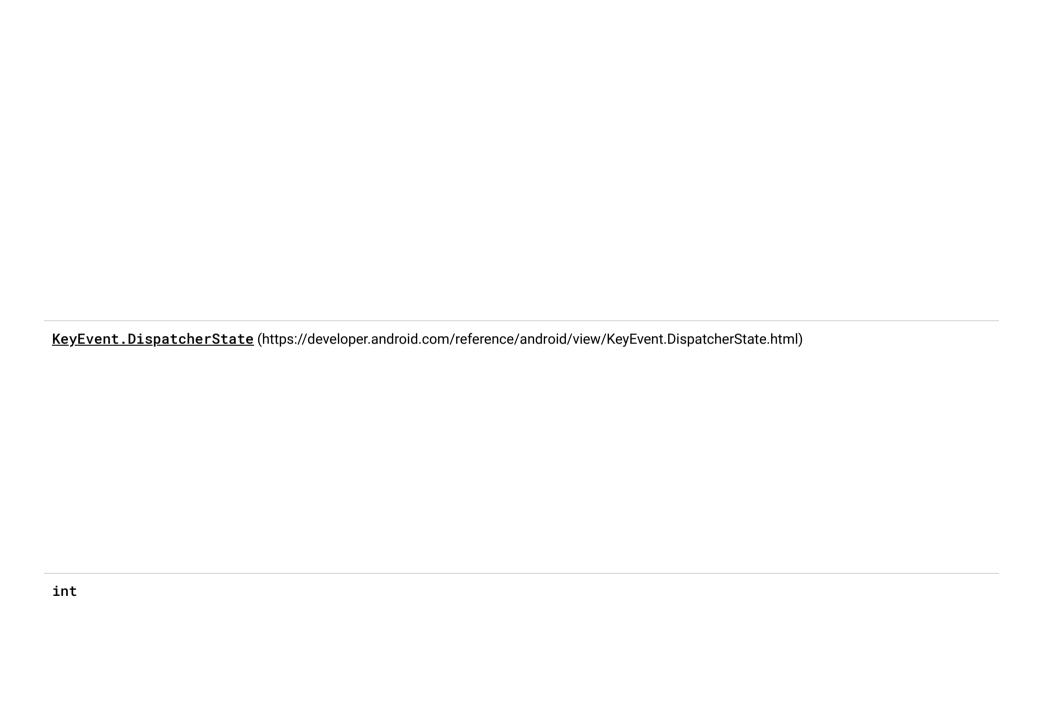
<u>Drawable</u> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)	
int	
<u>ColorStateList</u> (https://developer.android.com/reference/android/content/res/ColorStateList.html)	
<u>PorterDuff.Mode</u> (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)	

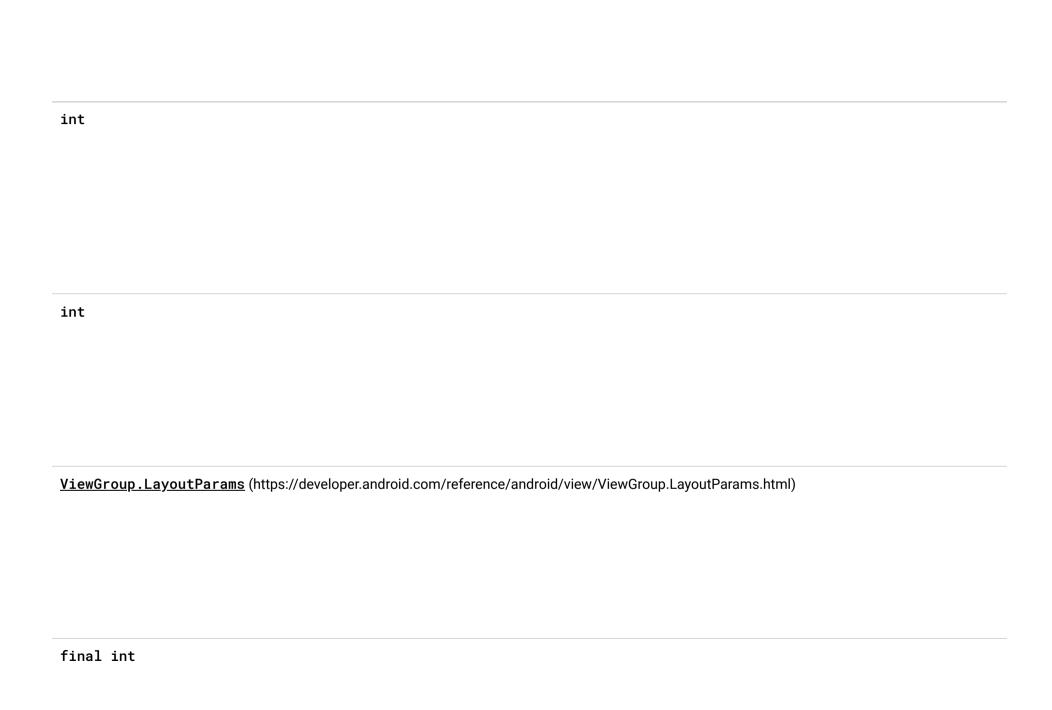


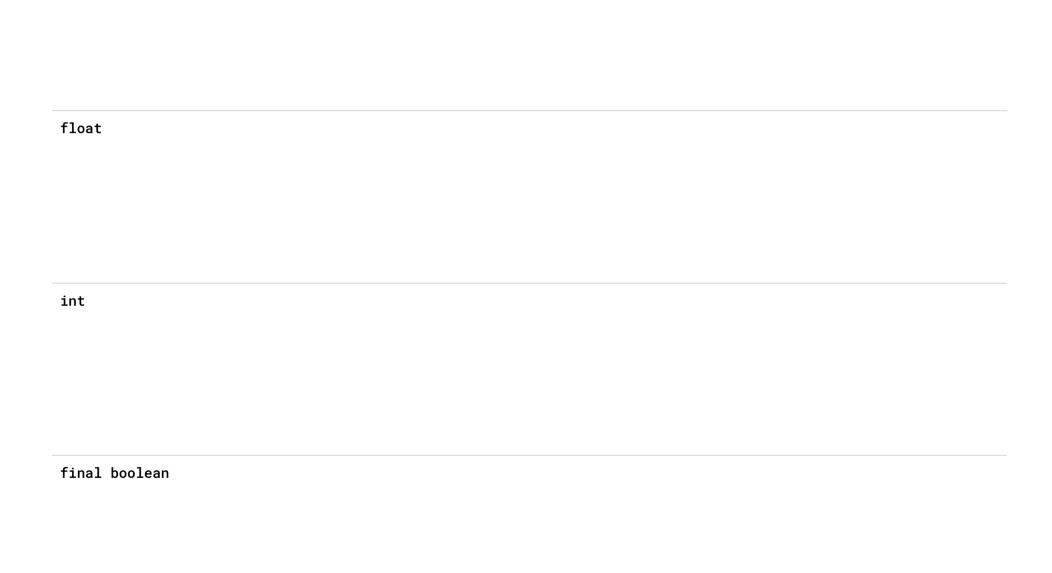
<u>Handler</u> (https://developer.android.com/reference/android/os/Handler.html)	
final boolean	
final int	

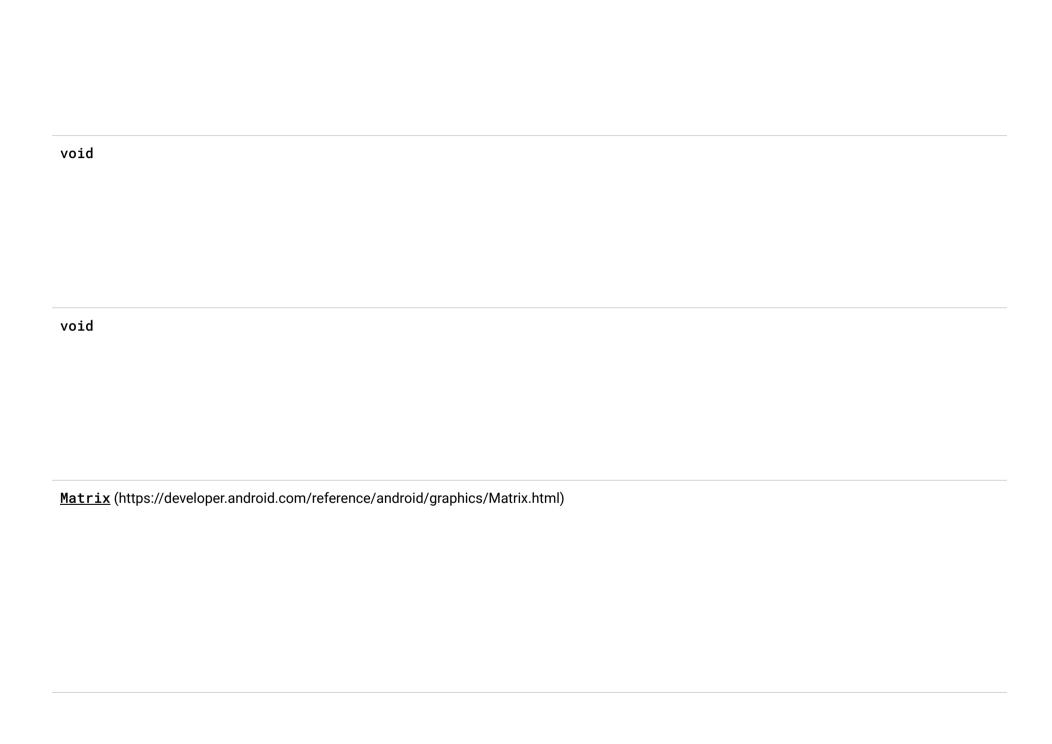












final int

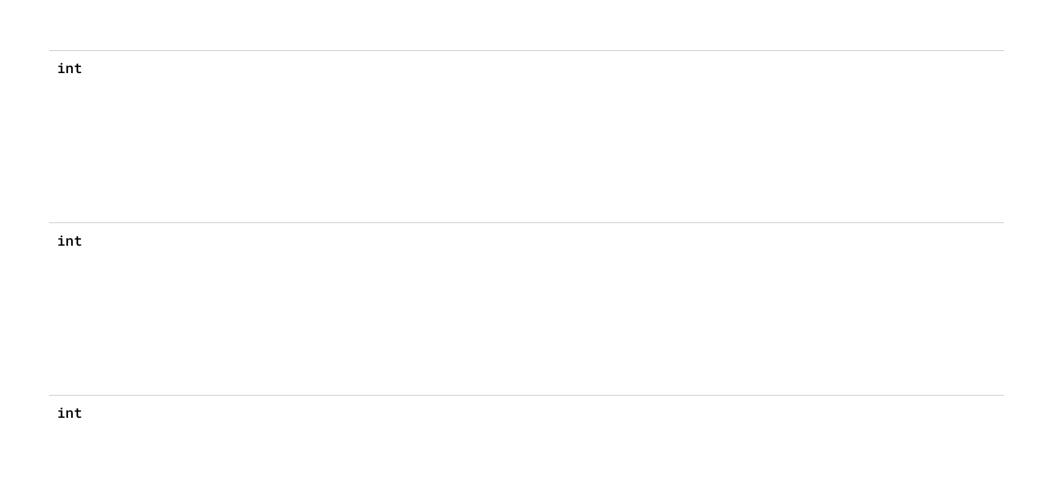
final int

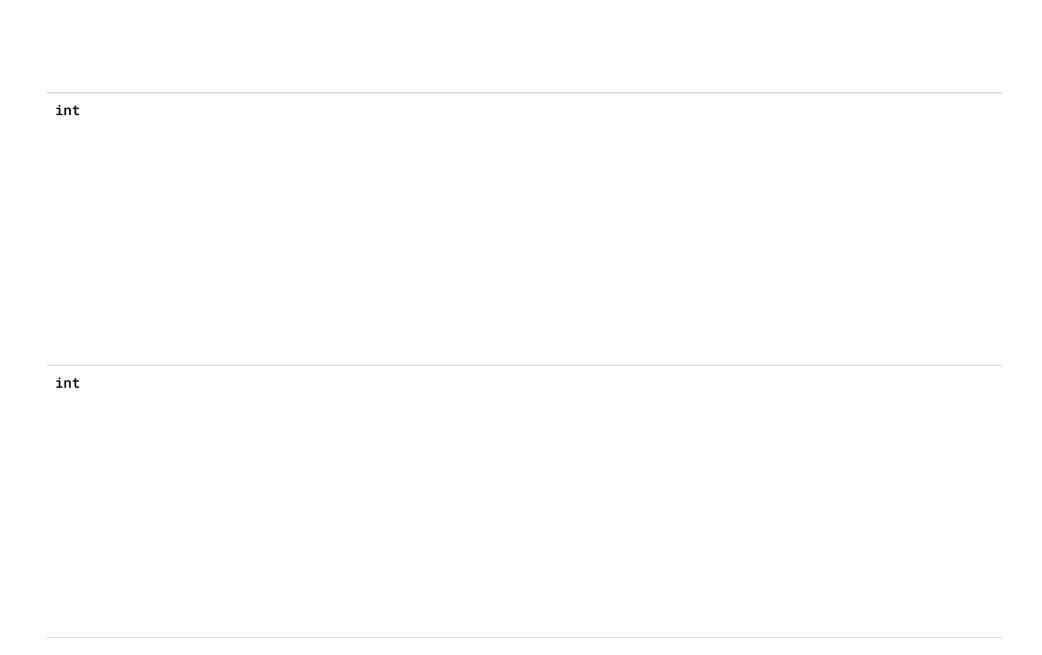
final int

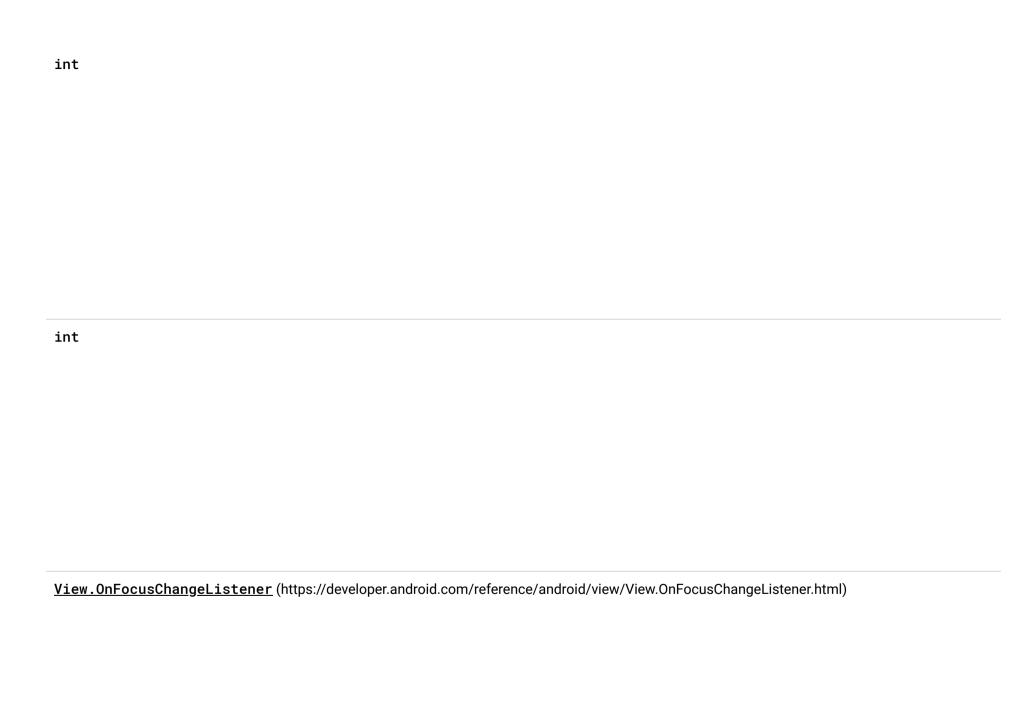
final int

final int

int

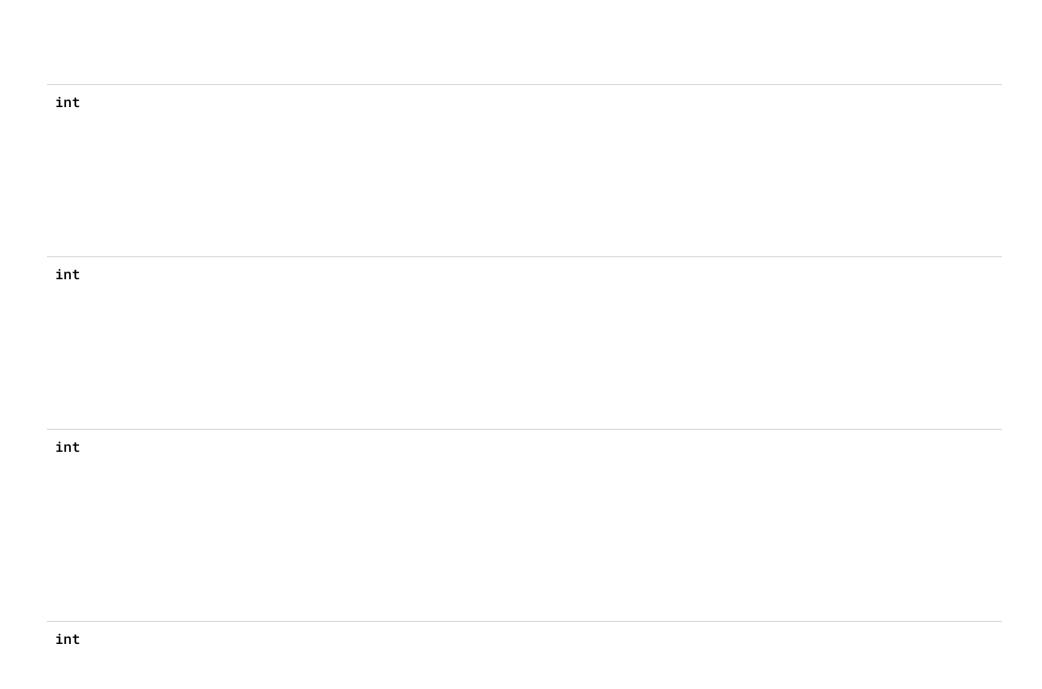


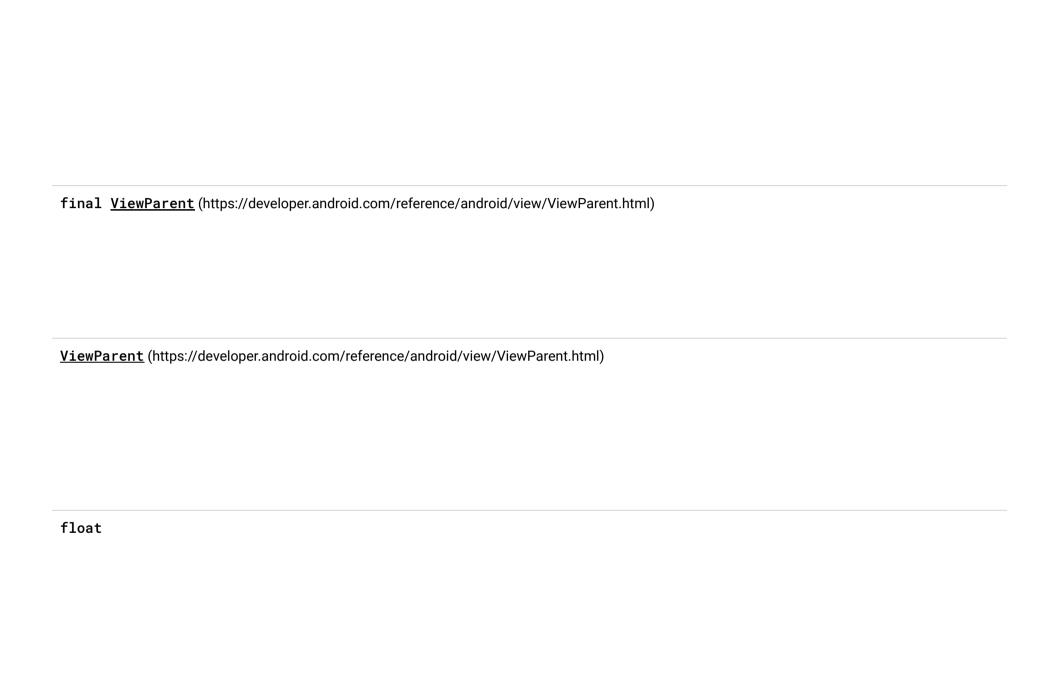


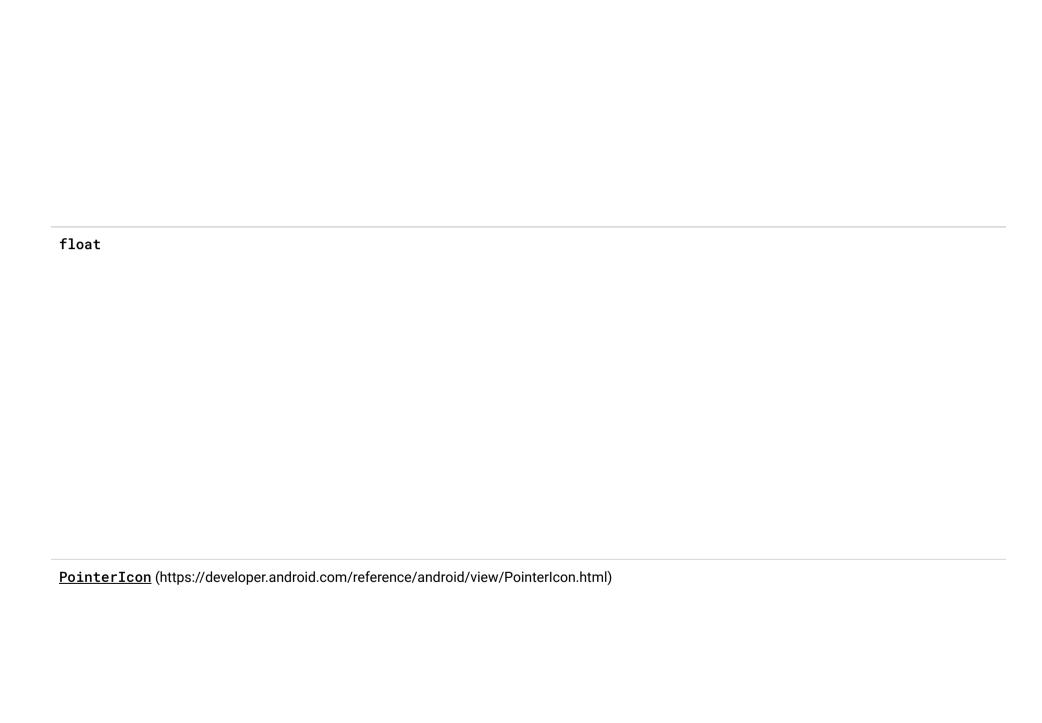


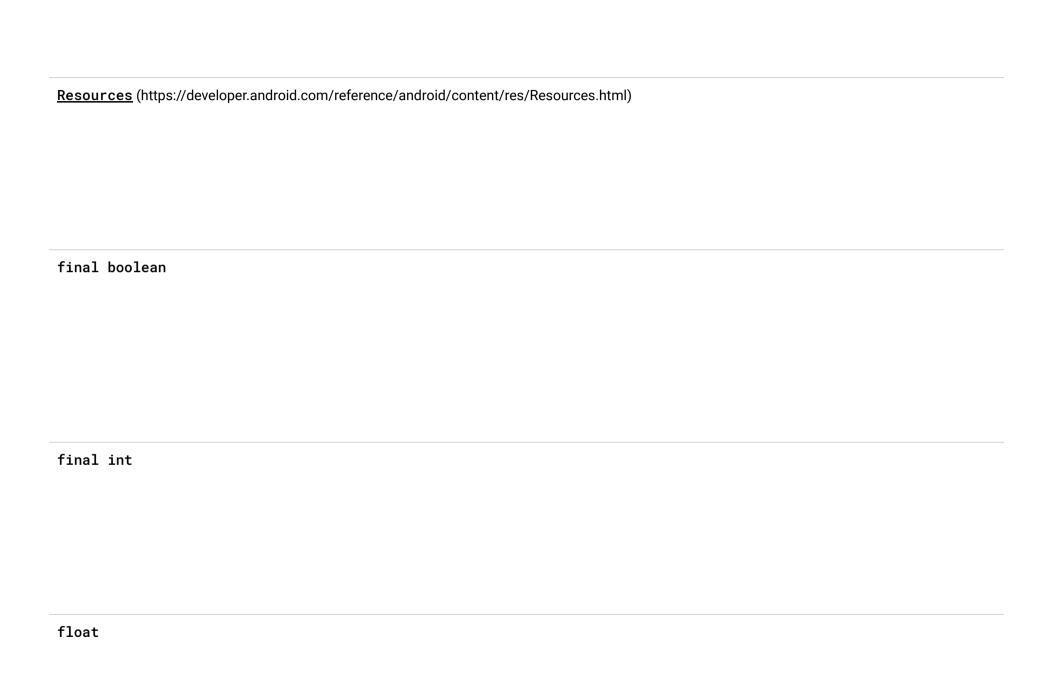
int	
<u>ViewOutlineProvider</u> (https://developer.android.com/reference/android/view/ViewOutlineProvider.html)	
int	
int	

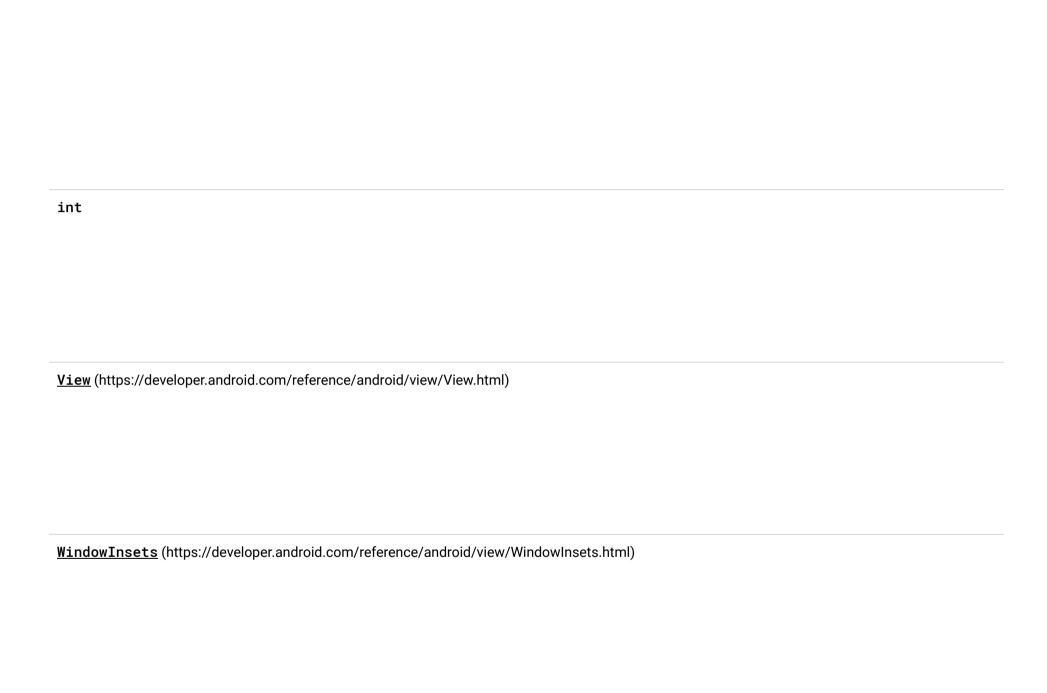
<u>ViewOverlay</u> (https://developer.android.com/reference/android/view/ViewOverlay.htm	nl)
int	
int	

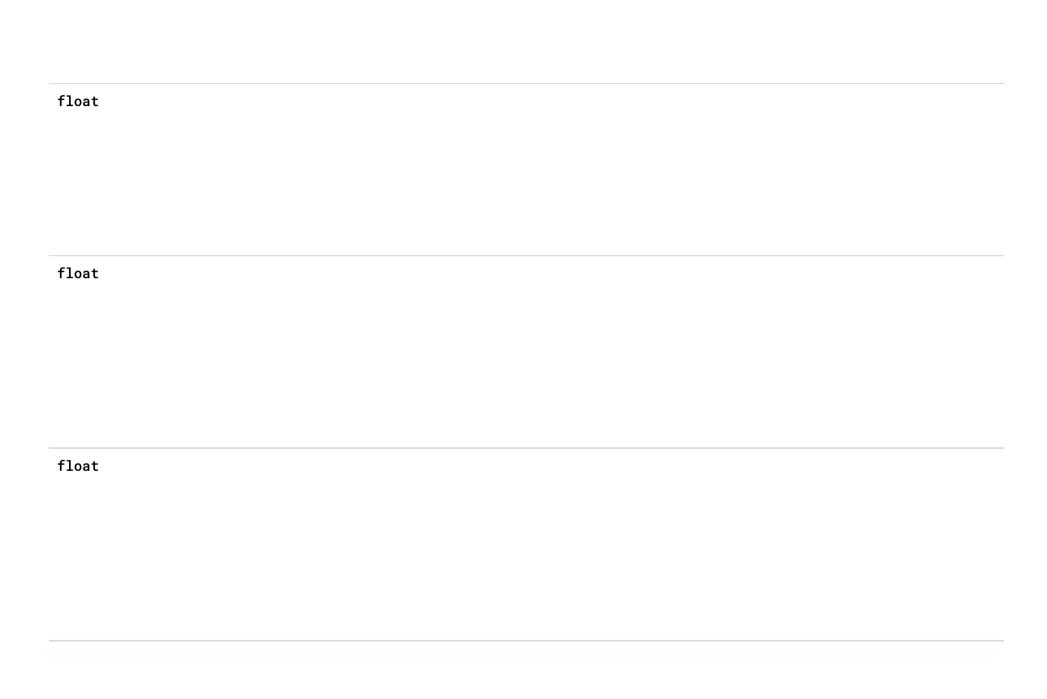


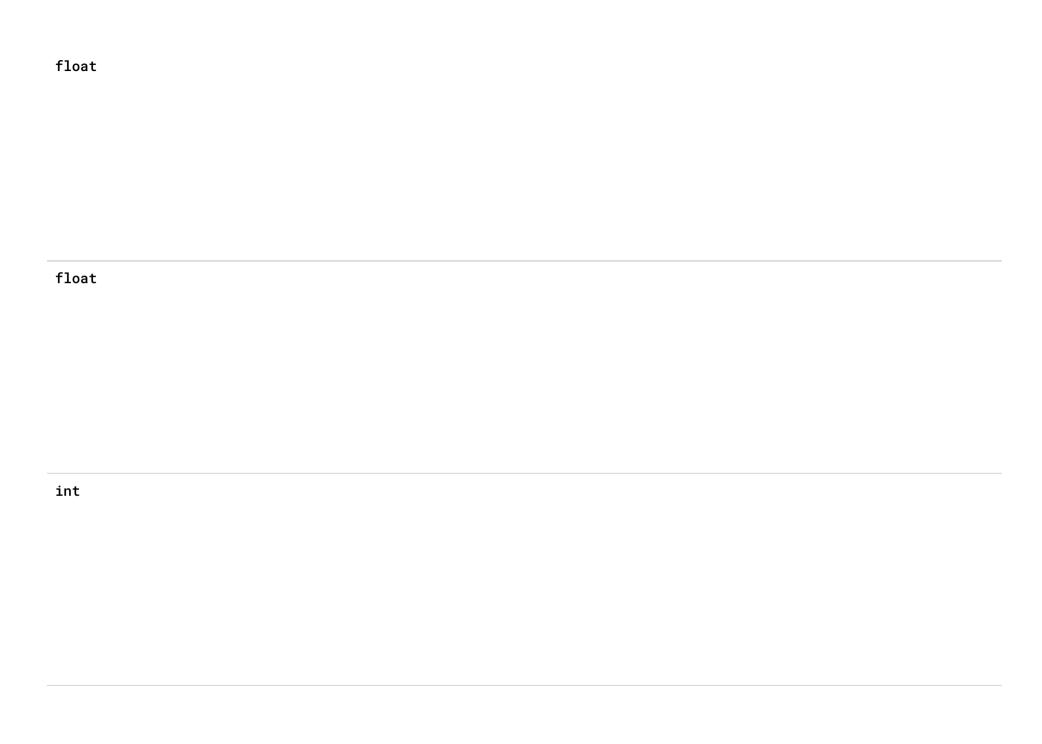




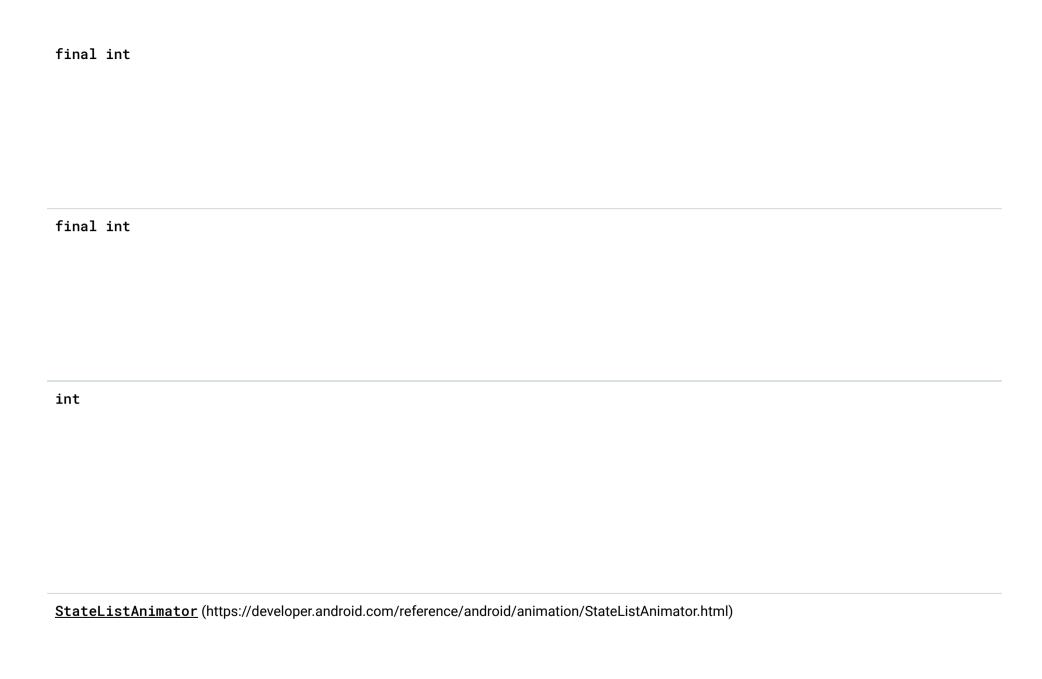


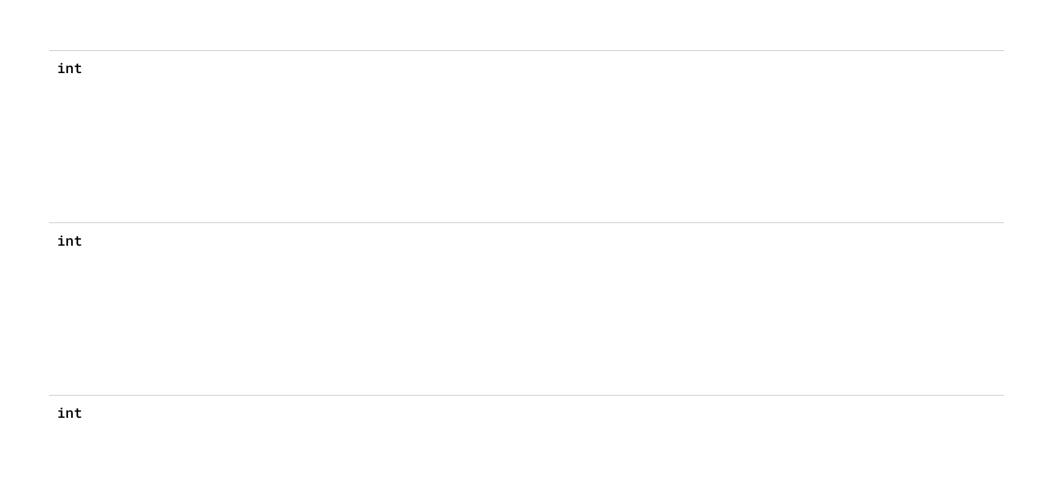






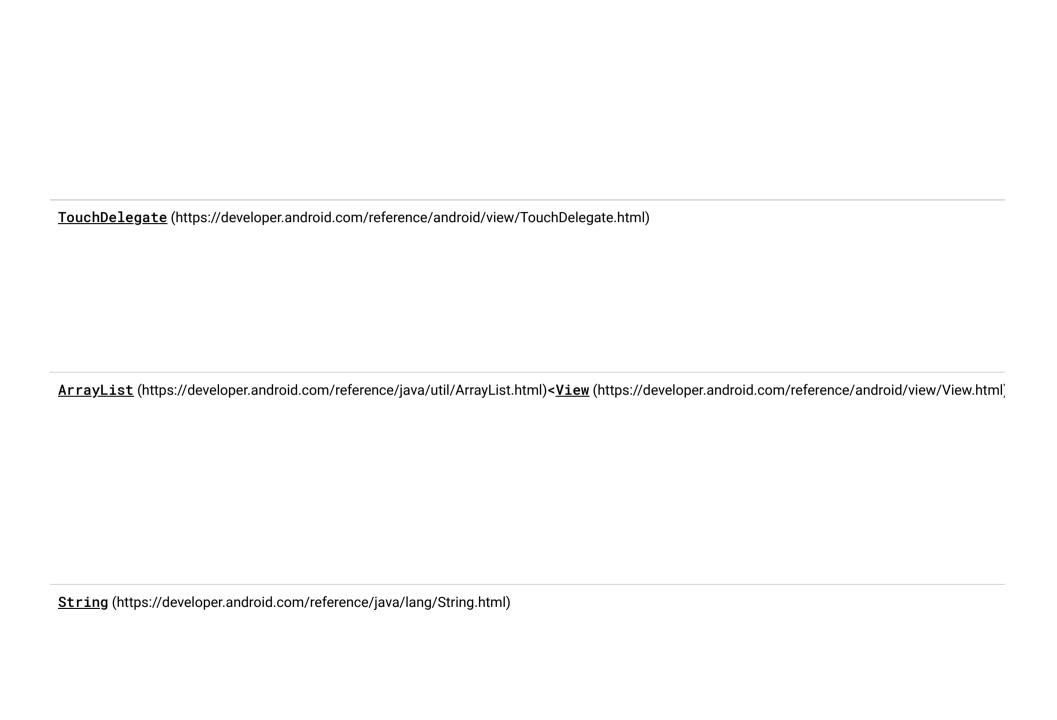
int		
int		
int		
1110		
int		

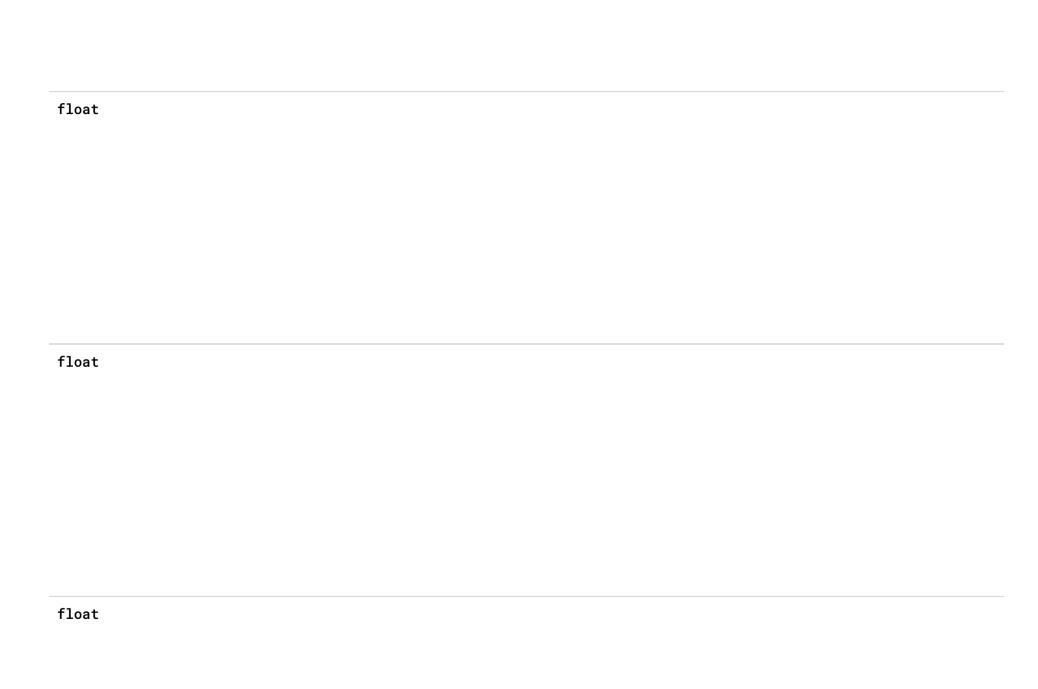


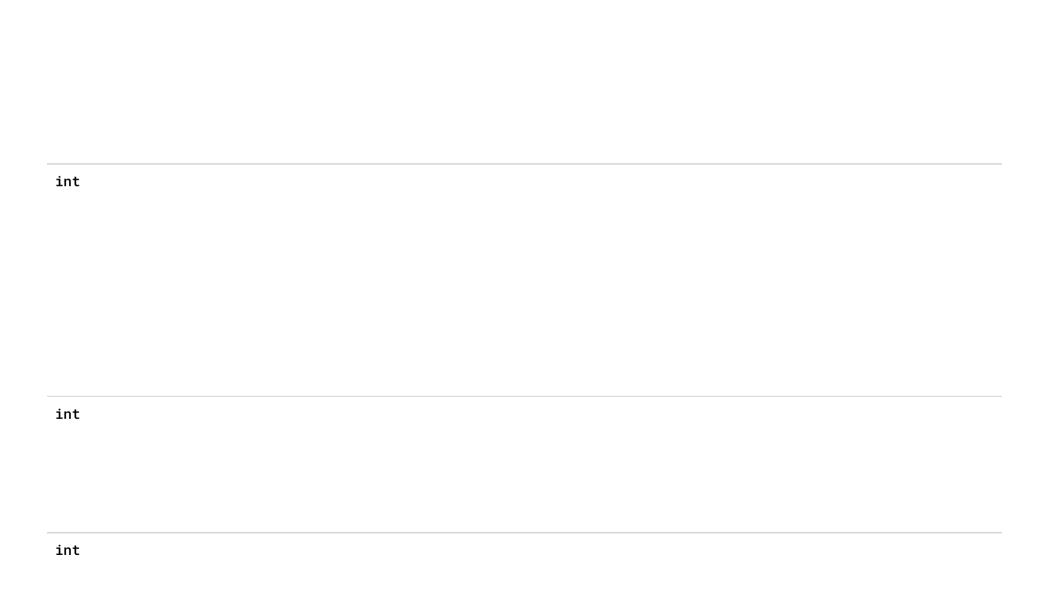


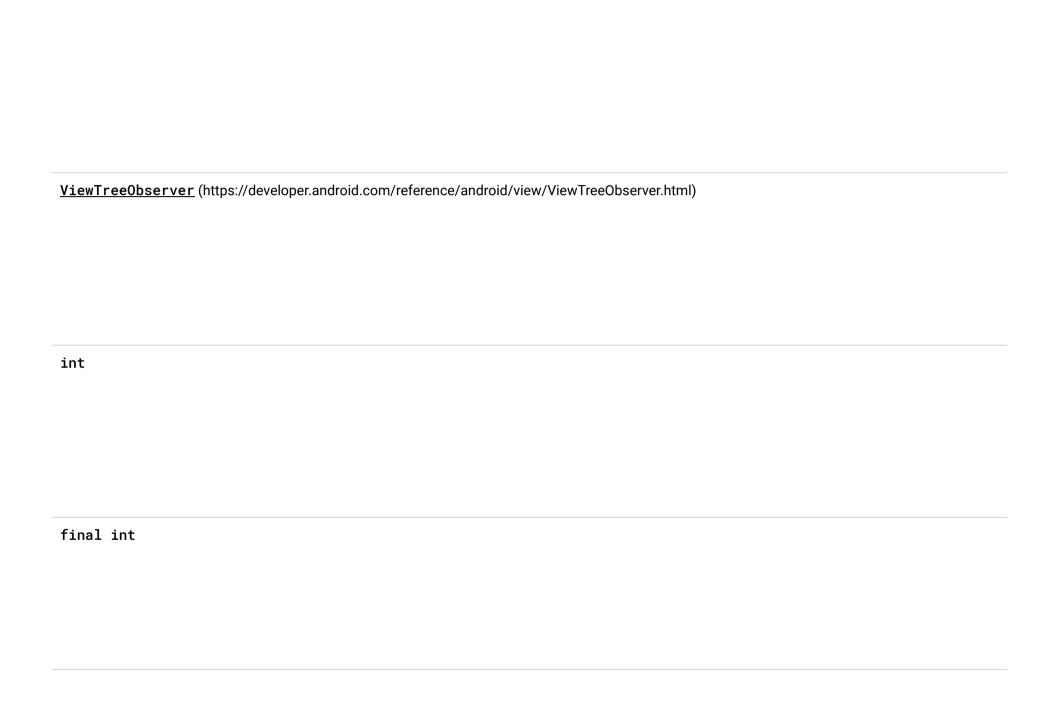
Object (https://developer.android.com/reference/java/lang/Object.html)
<u>Object</u> (https://developer.android.com/reference/java/lang/Object.html)
int
int

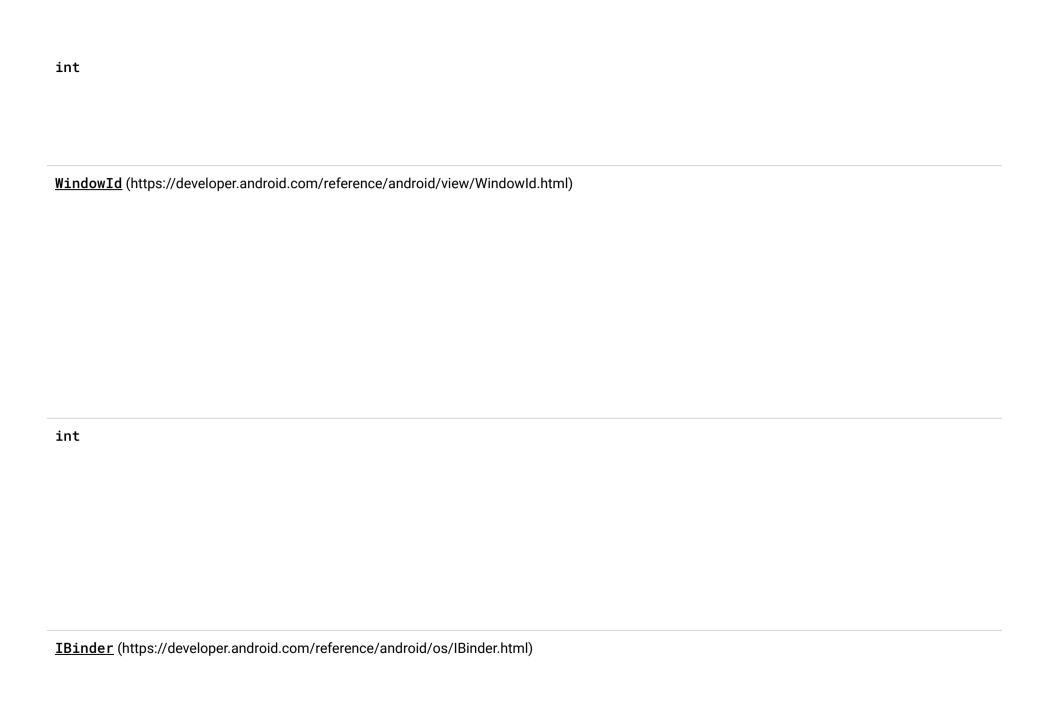
<u>CharSequence</u> (https://developer.android.com/reference/java/lang/CharSequence.html)				
final int				
float				
int				



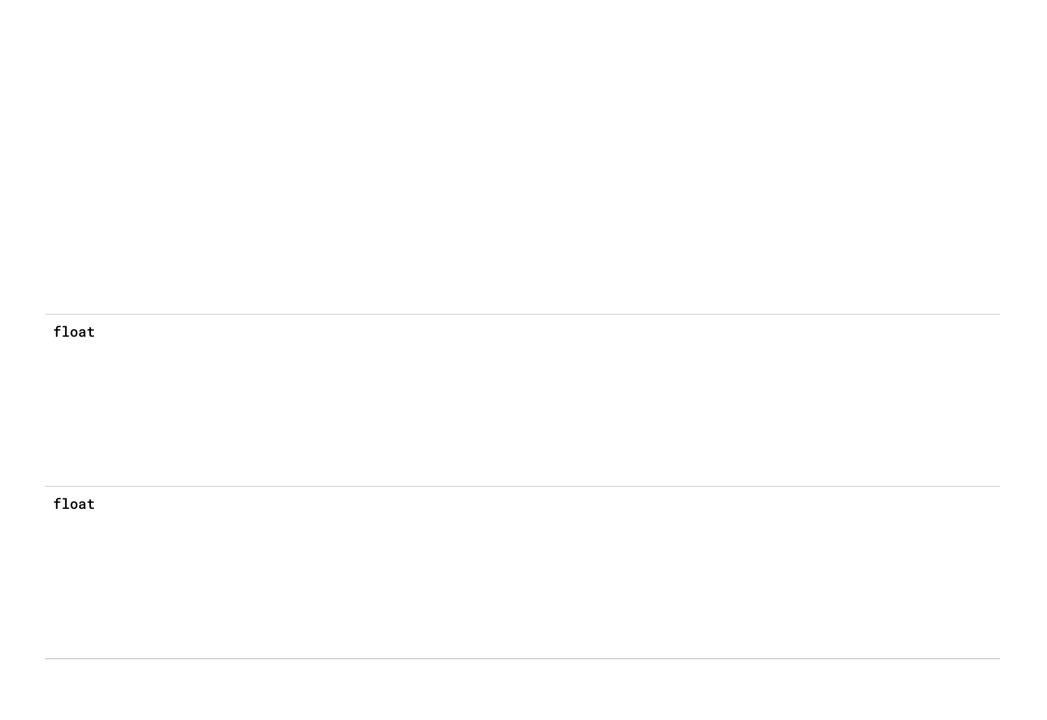


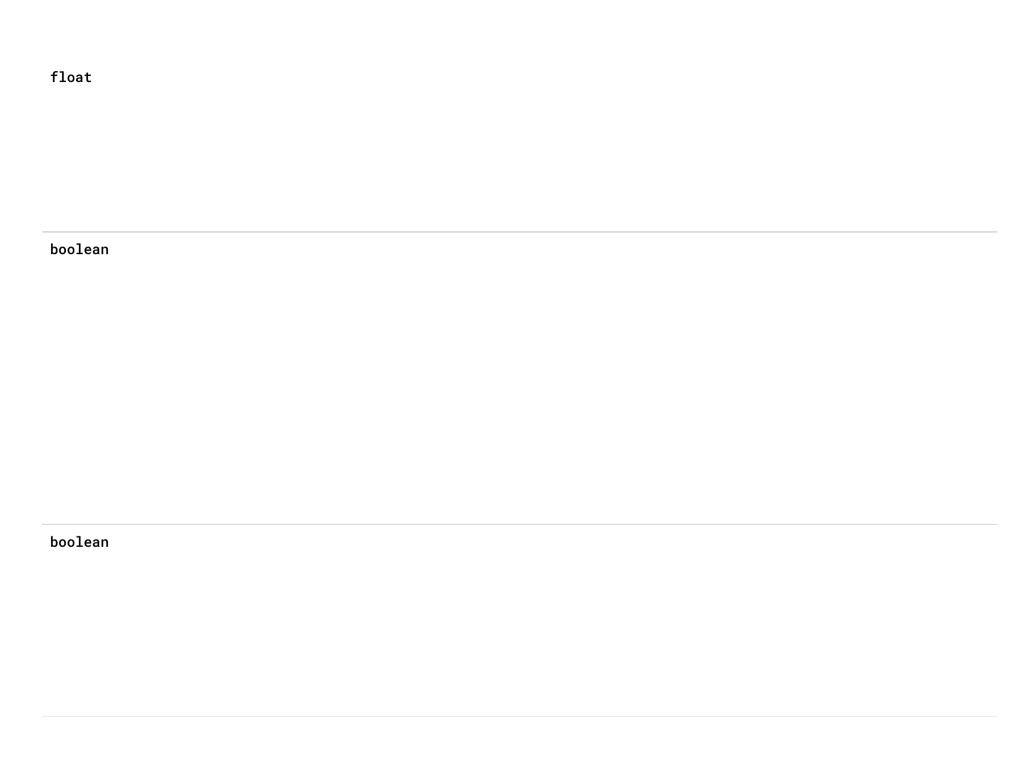


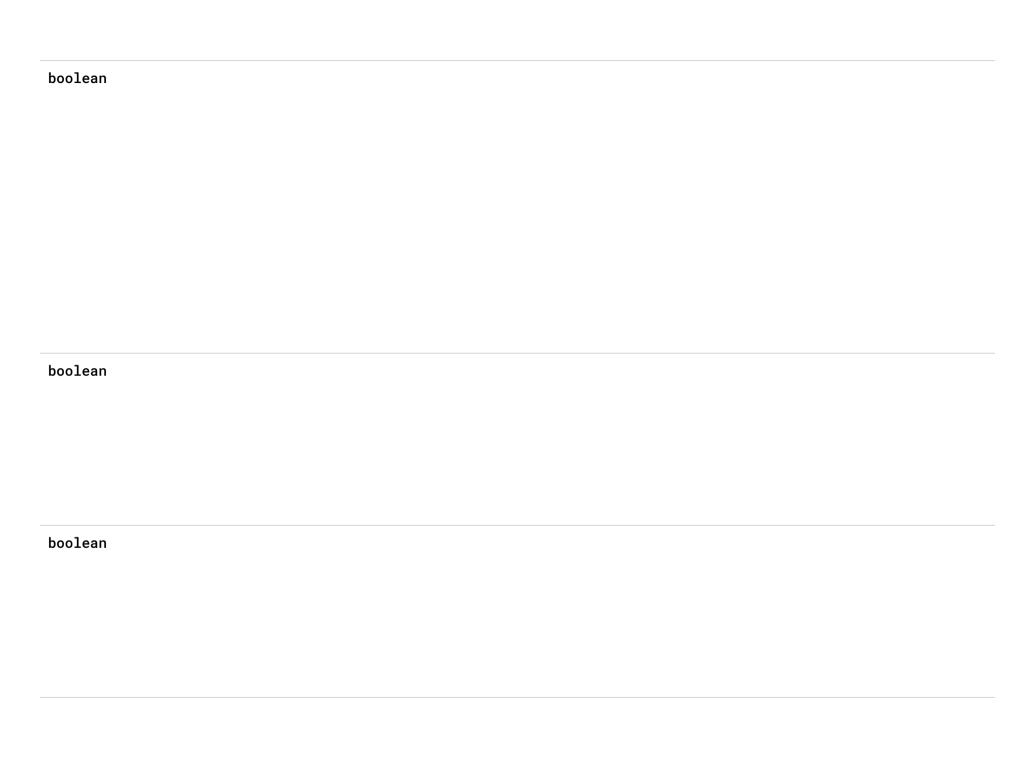


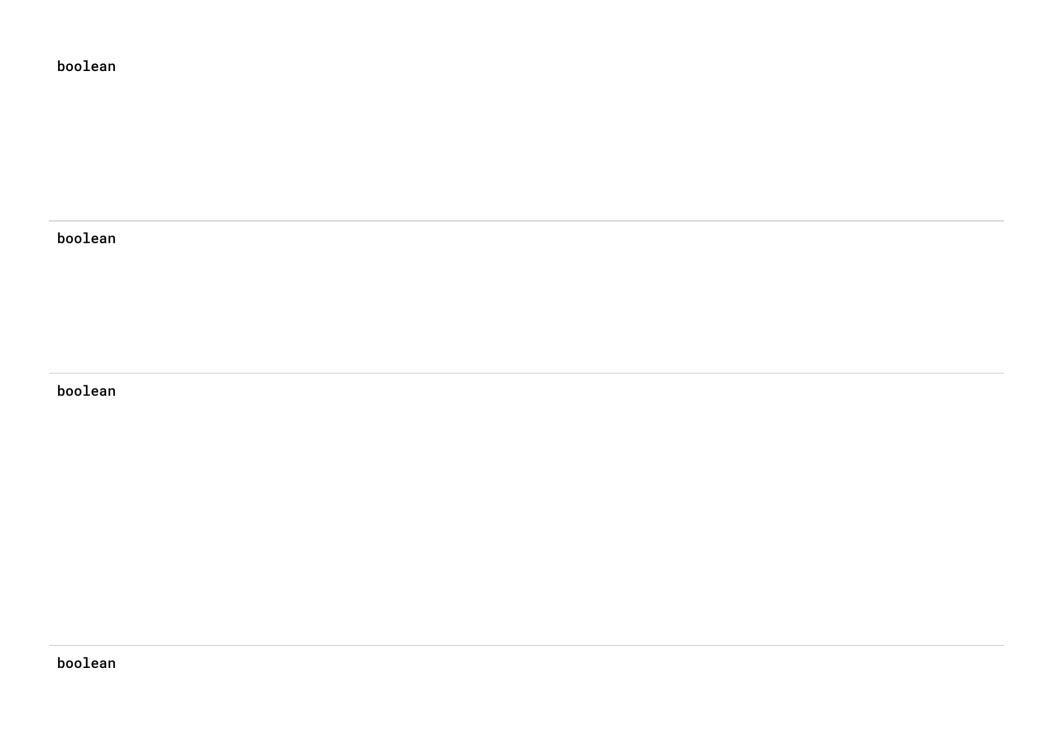


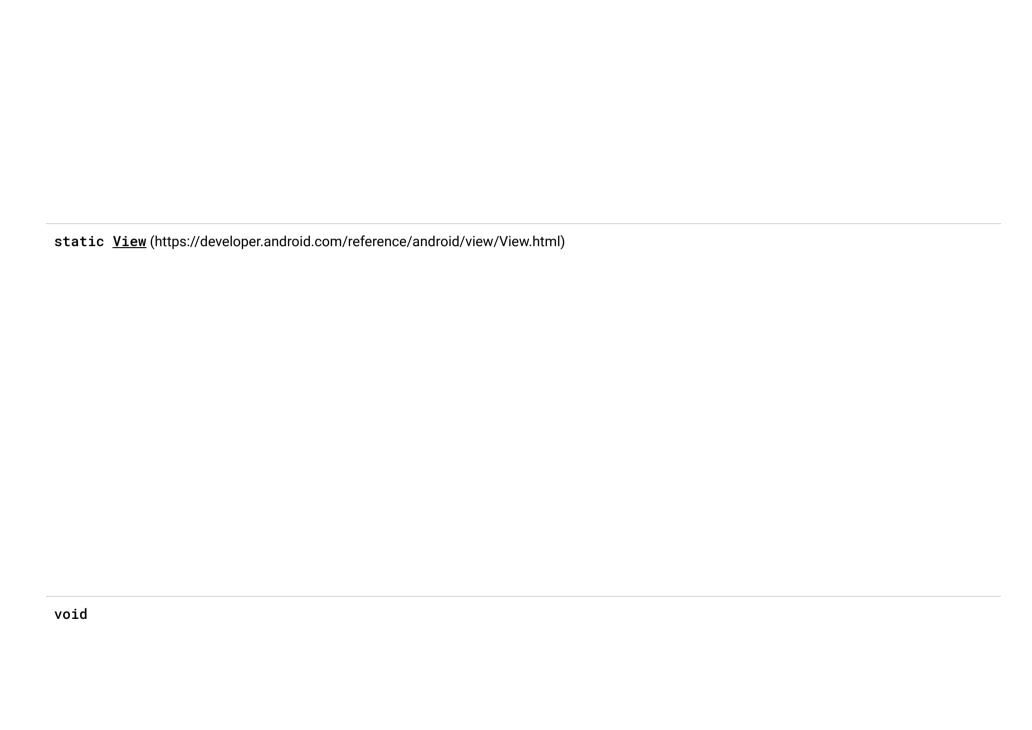






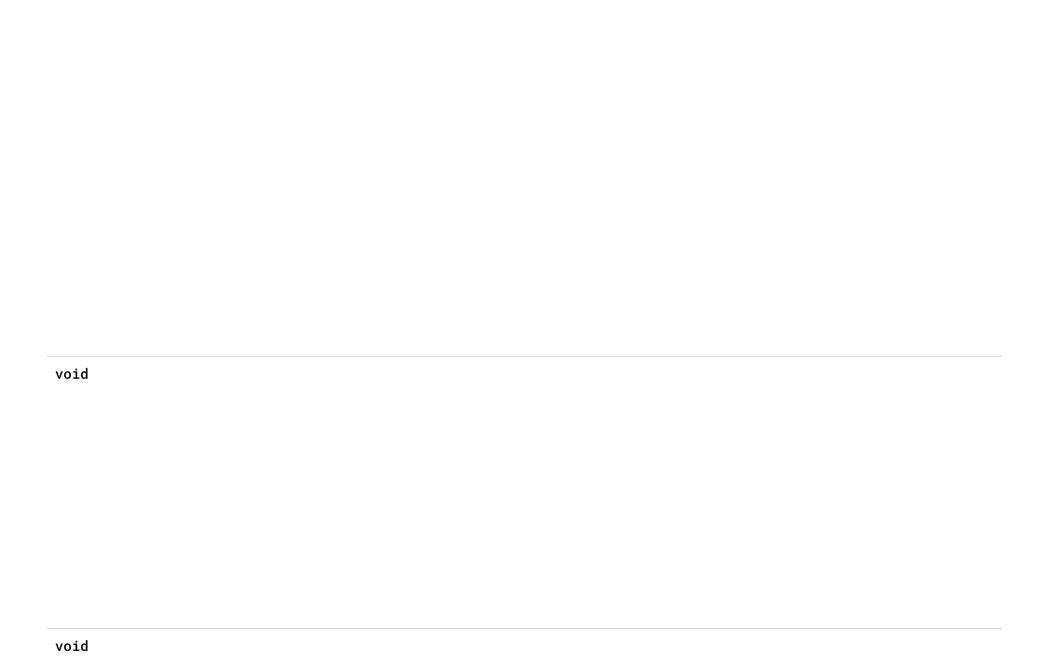


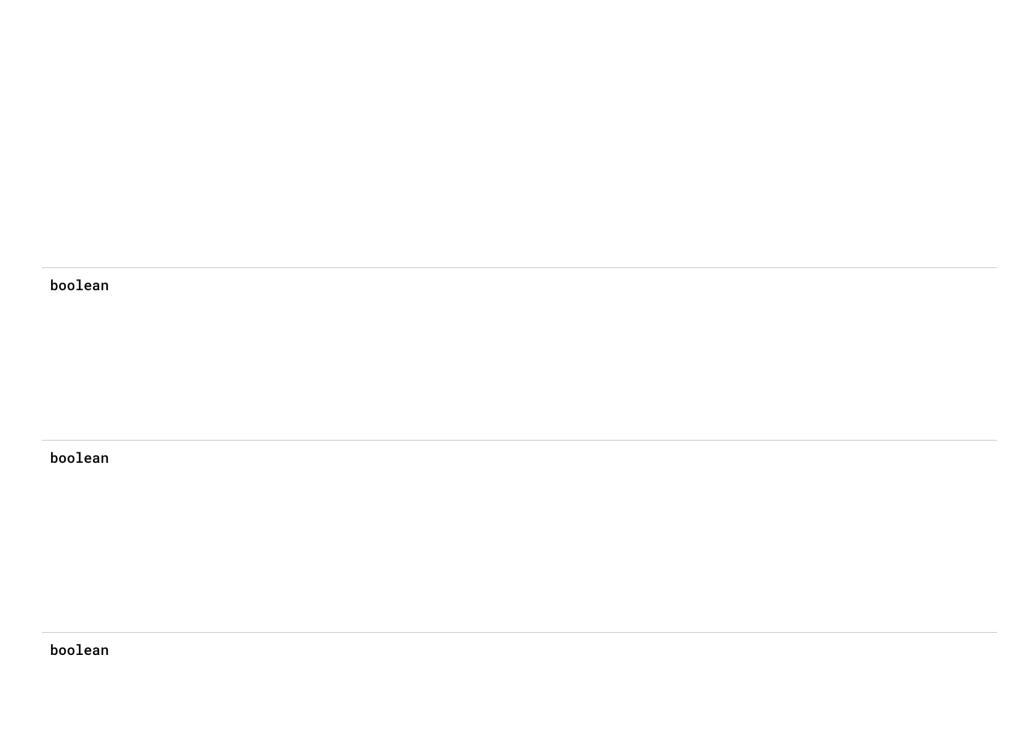


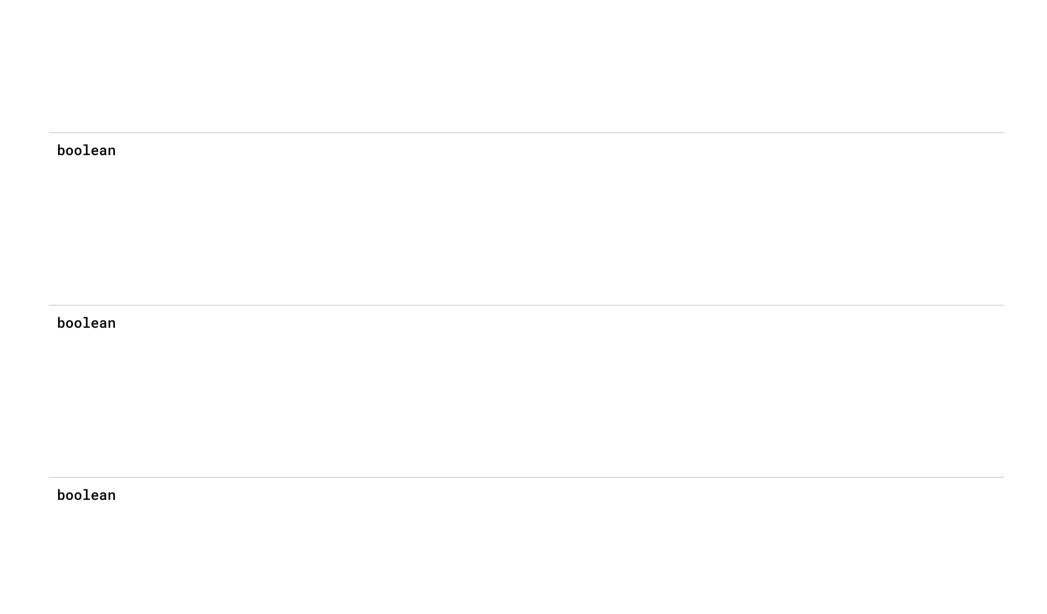


void

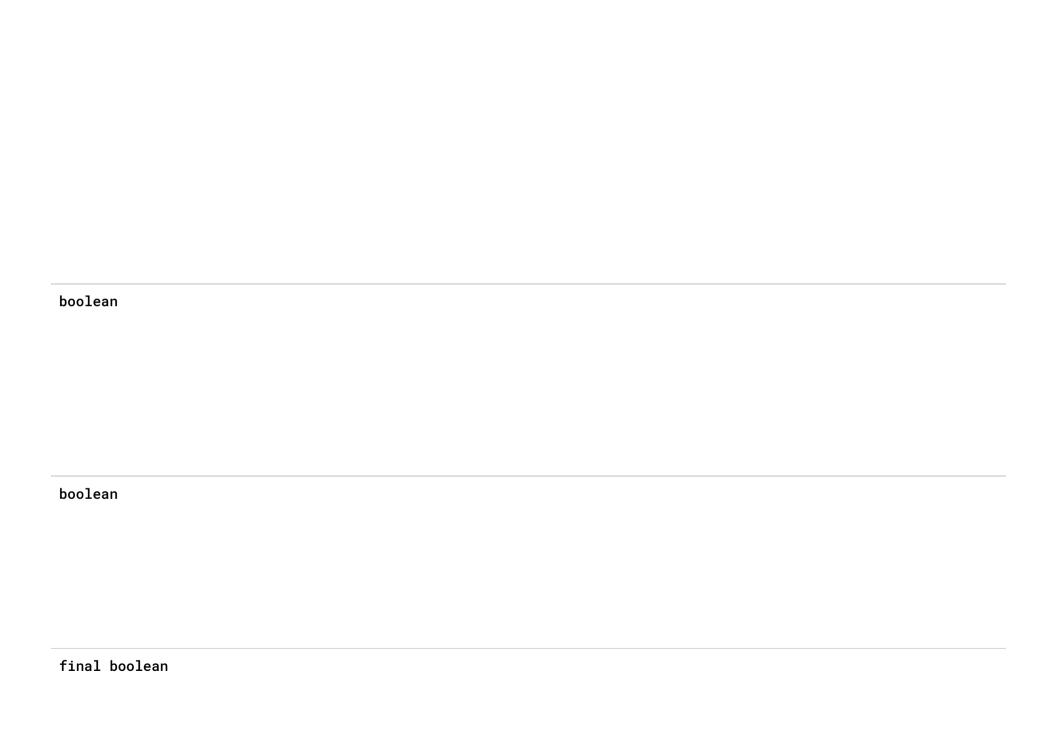
void

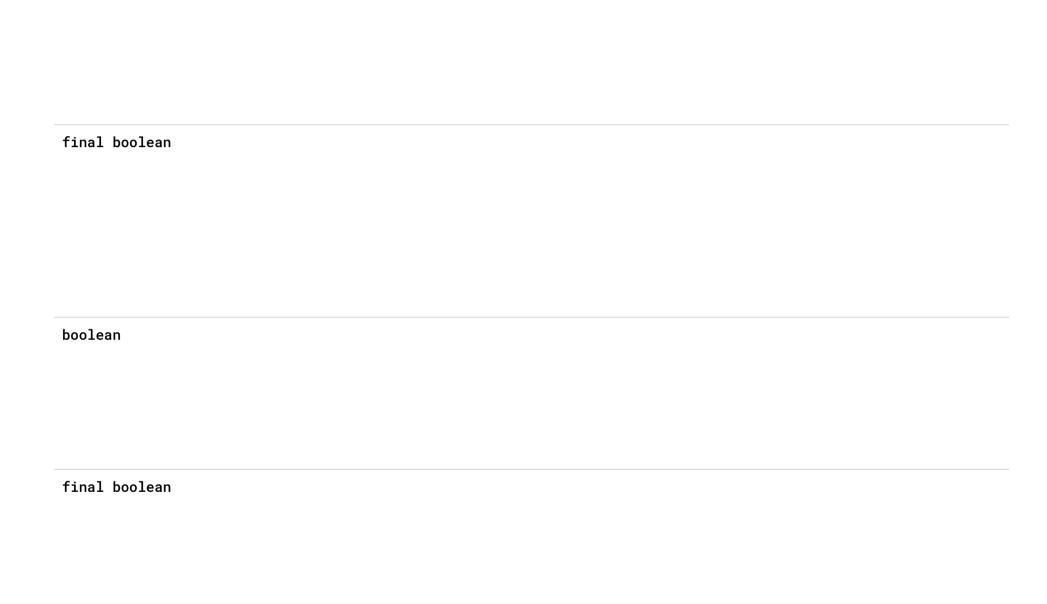




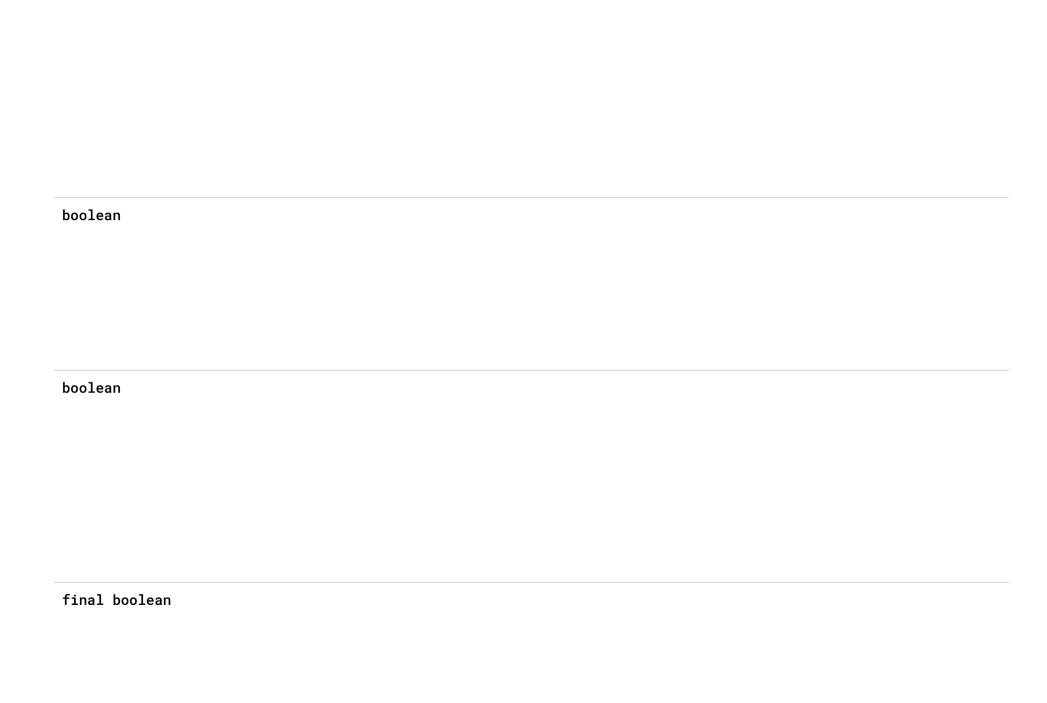


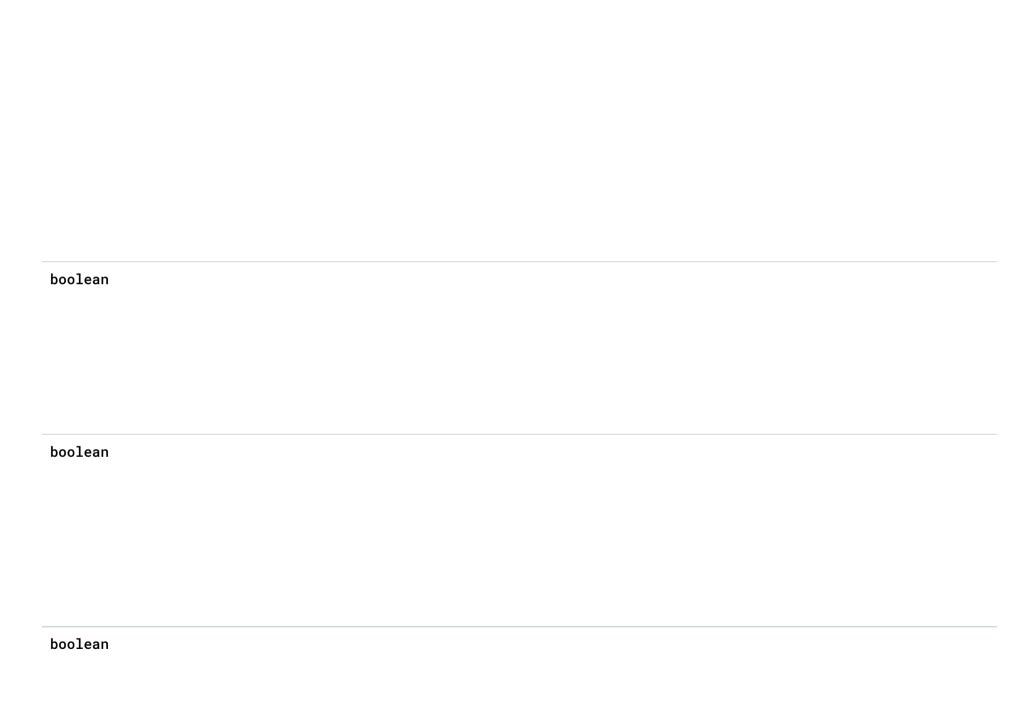
boolean

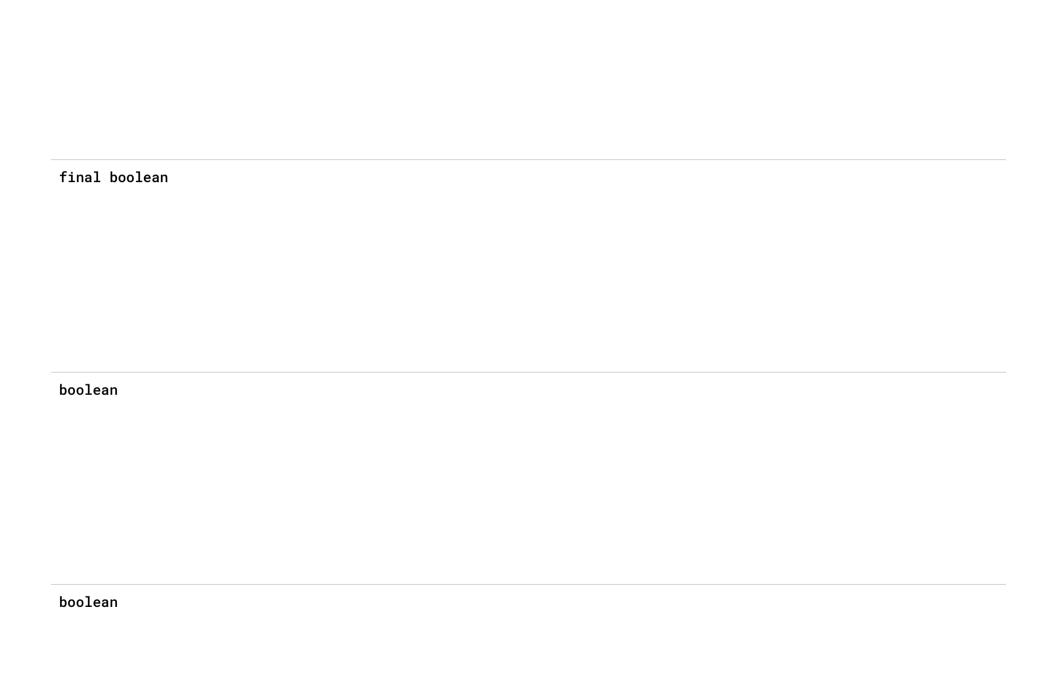


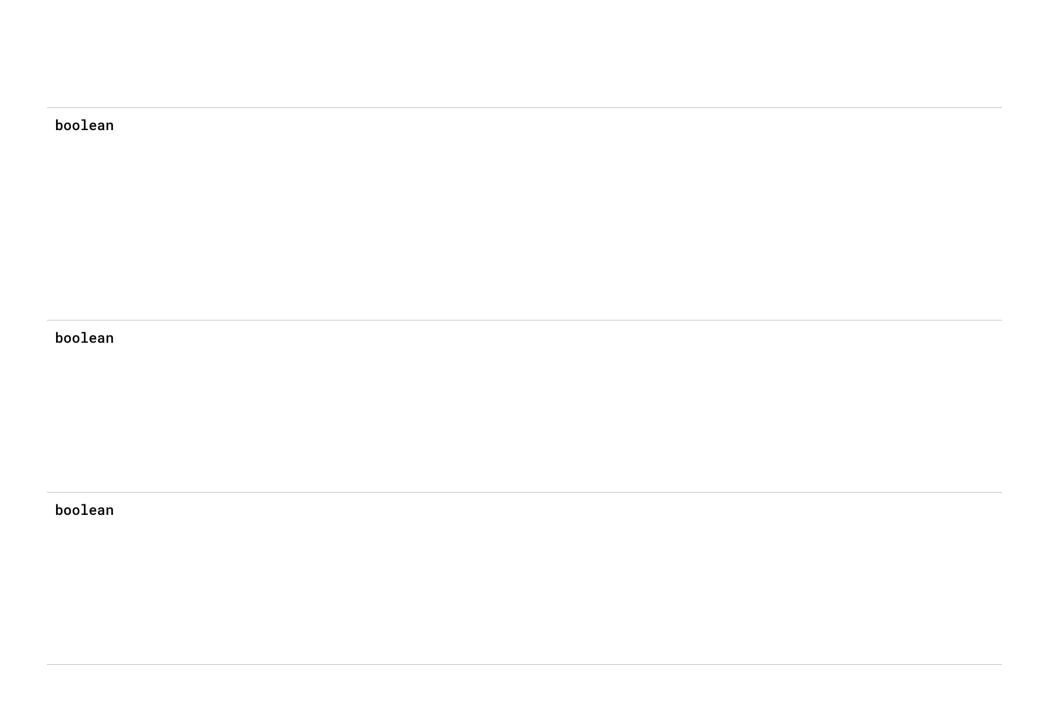


boolean		
boolean		
boolean		
Doolean		
boolean		

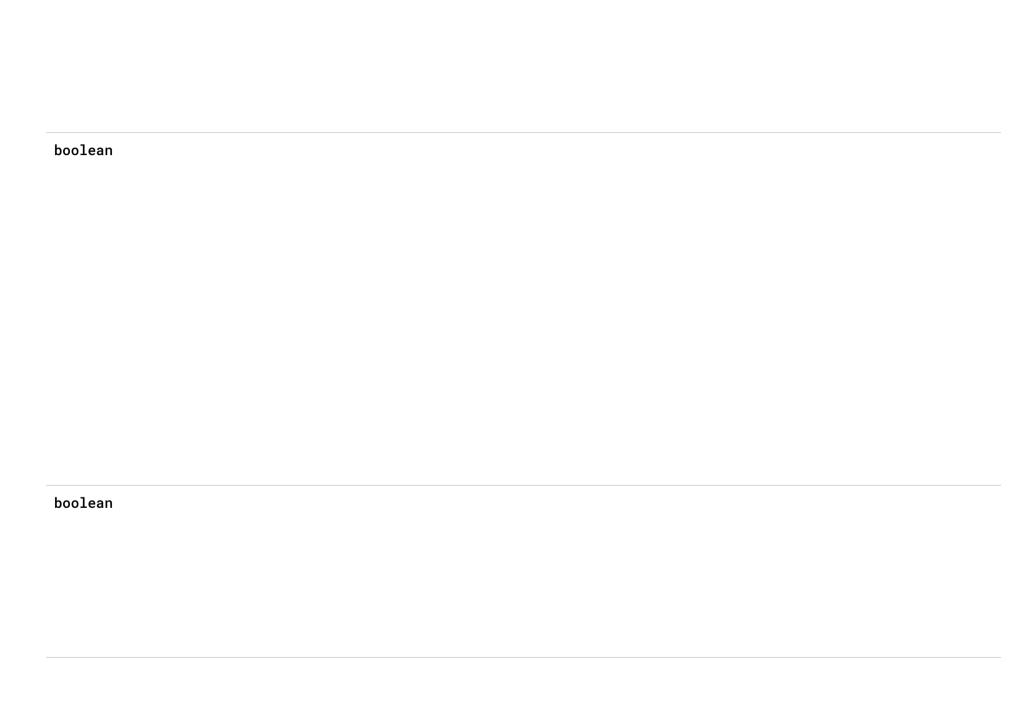


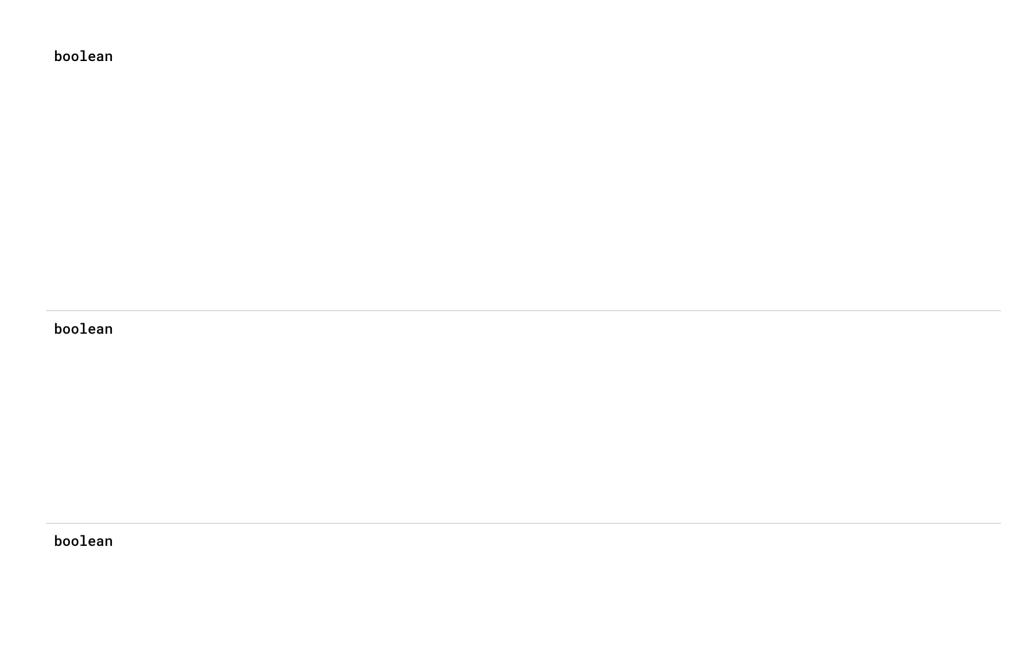






boolean		
boolean		
boolean		

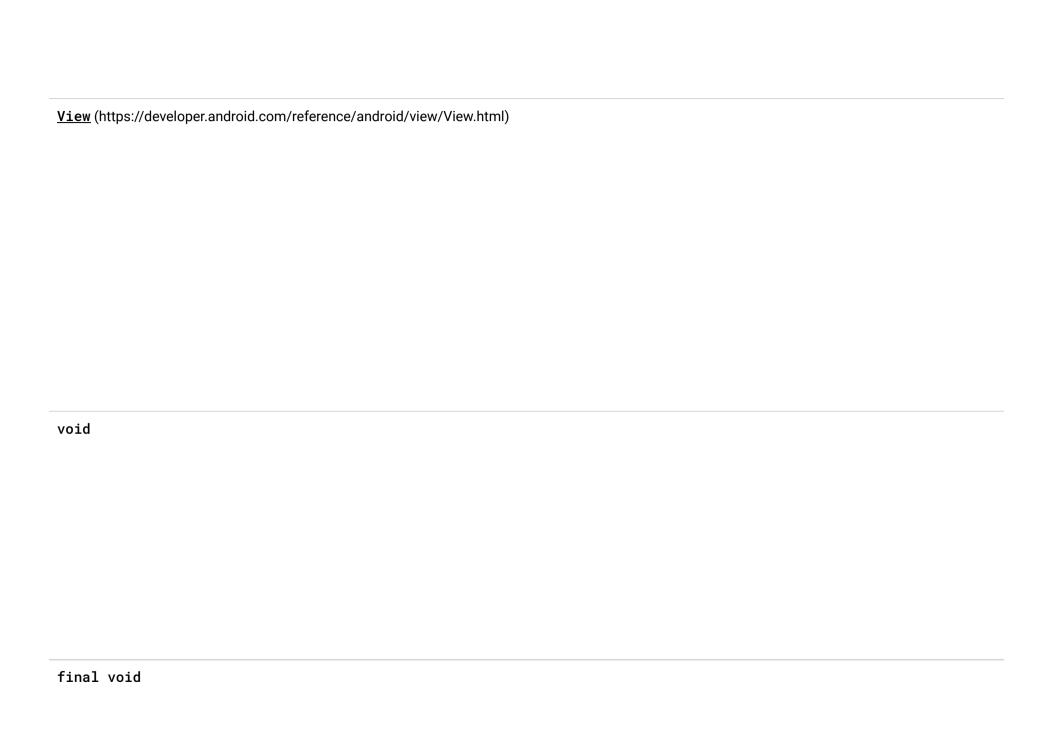




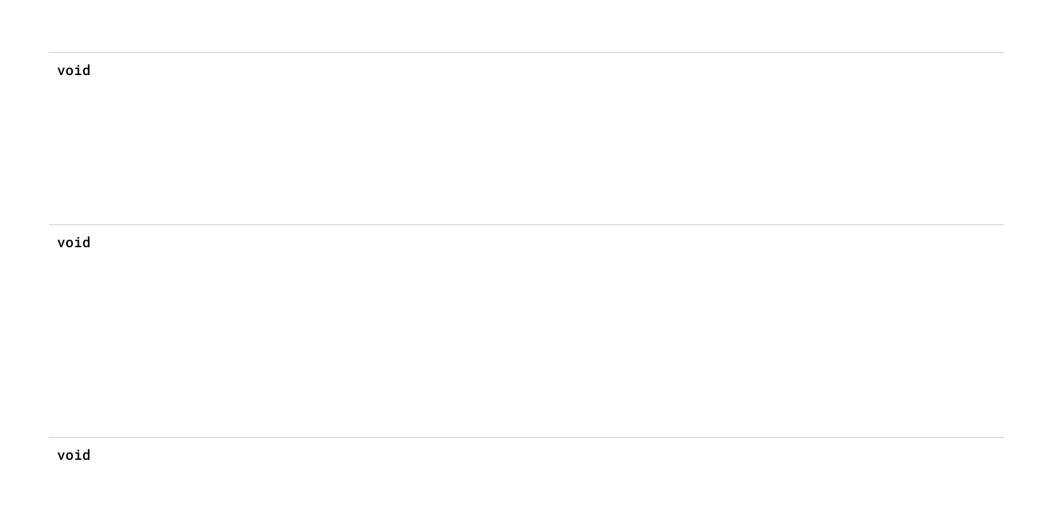
boolean		
boolean		
boolean		
boolean		

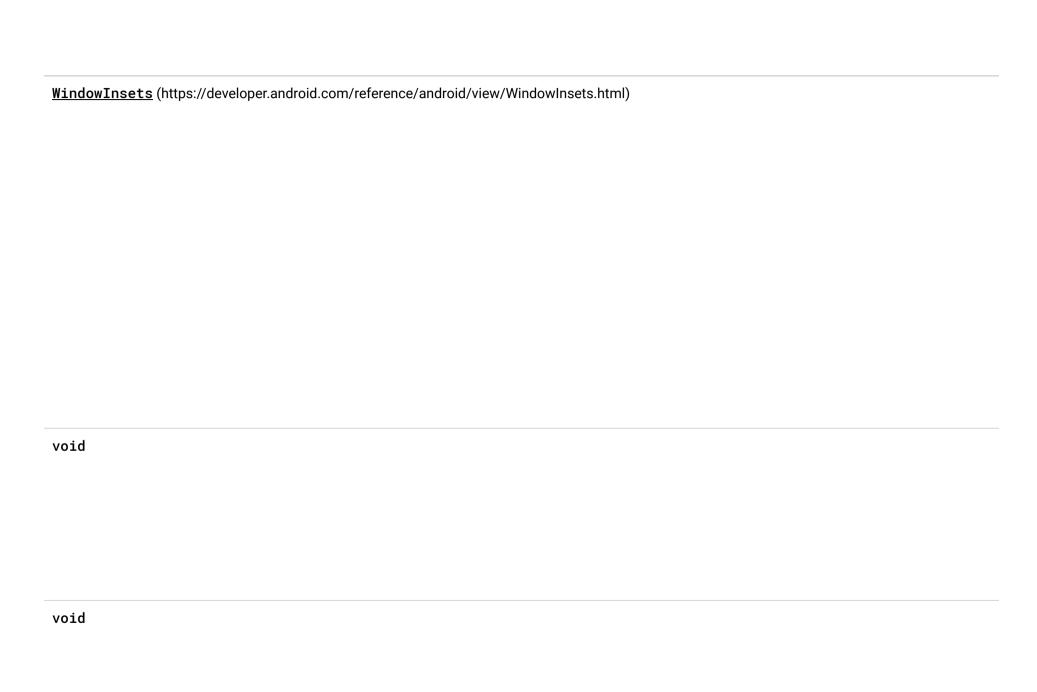
final boolean

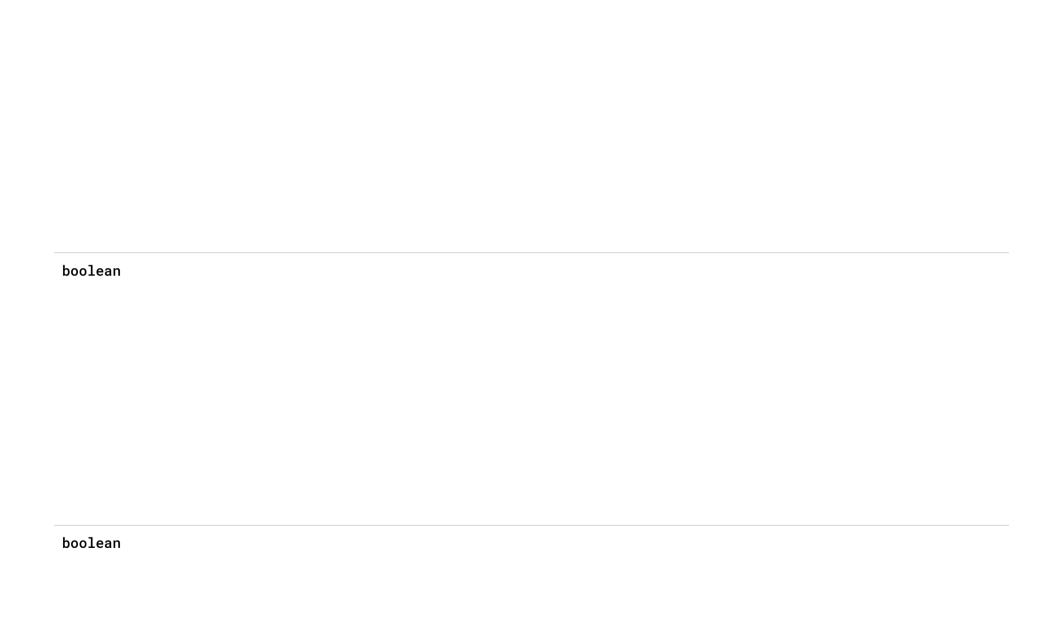
boolean		
boolean		
boolean		
boolean		



static int[]

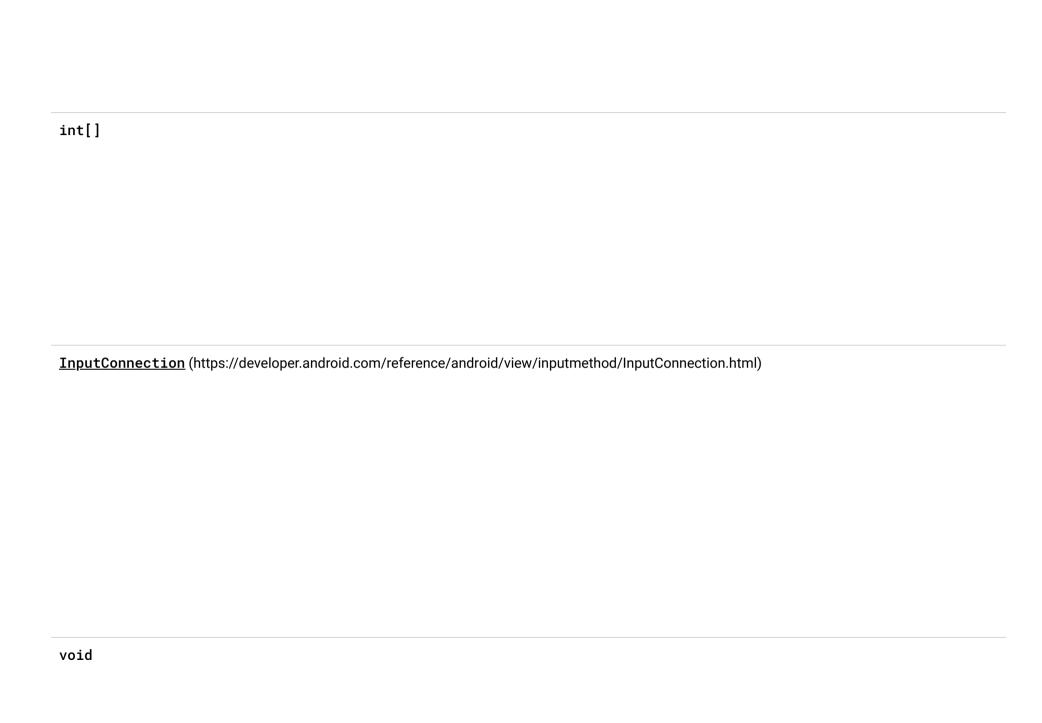


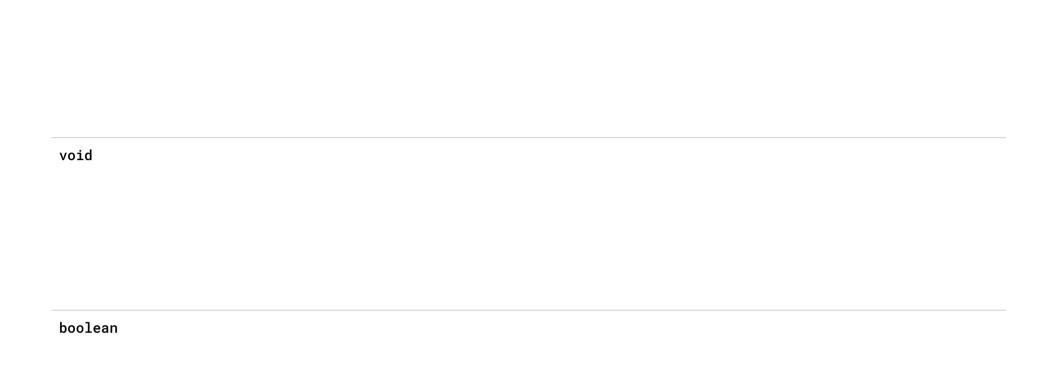




void

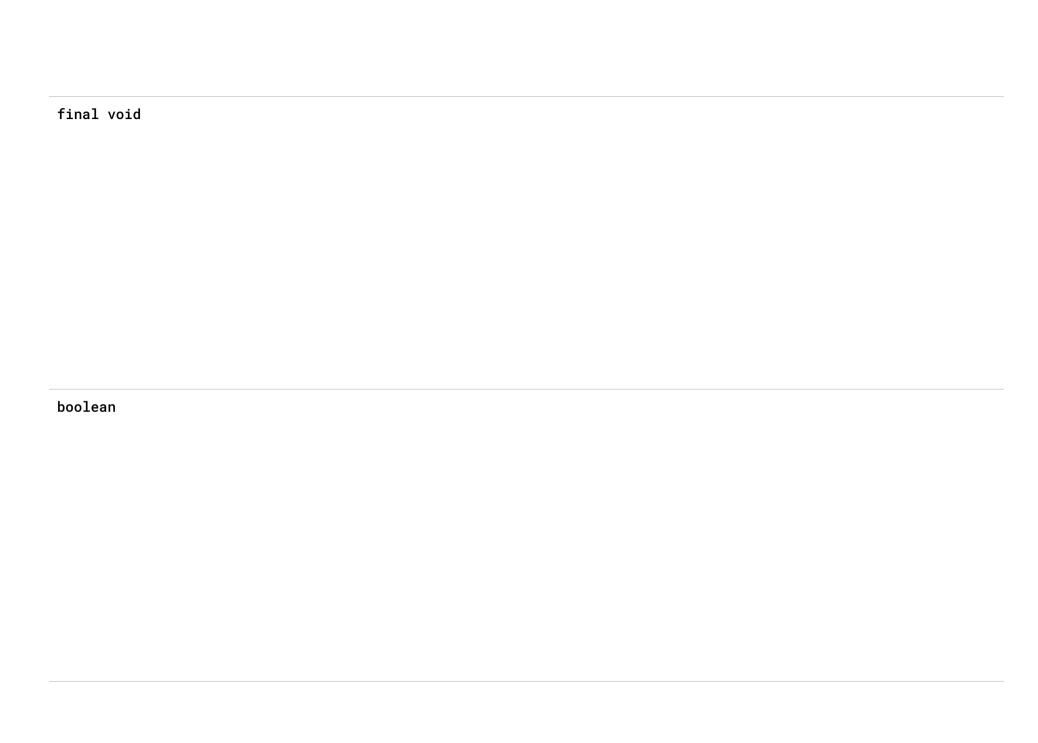
void

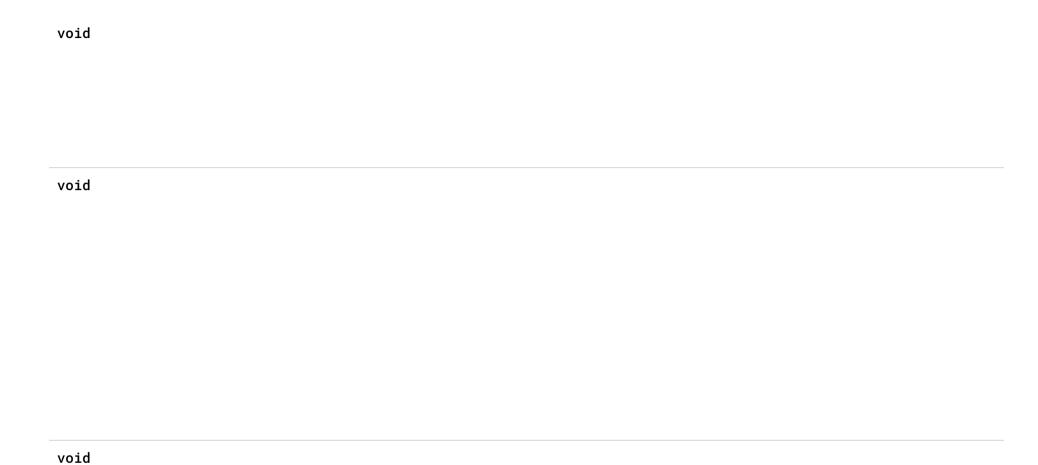


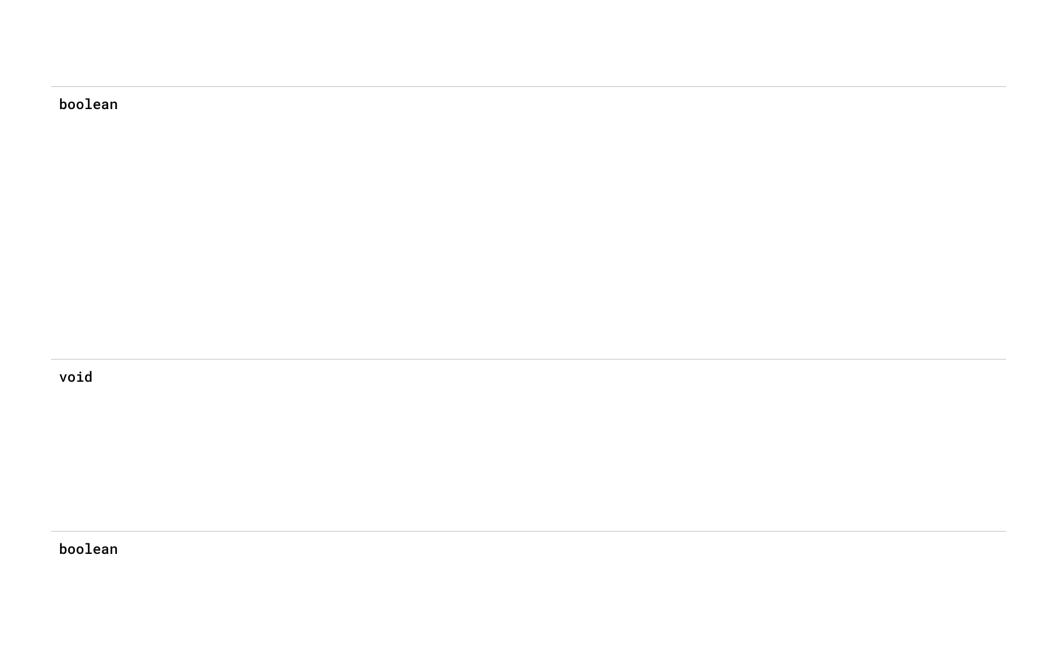


void

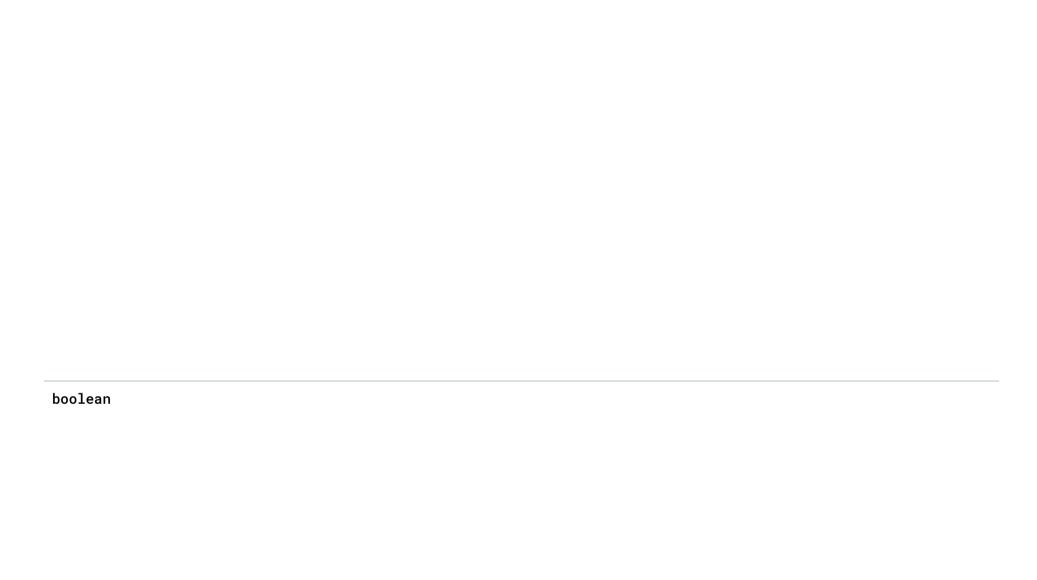
void

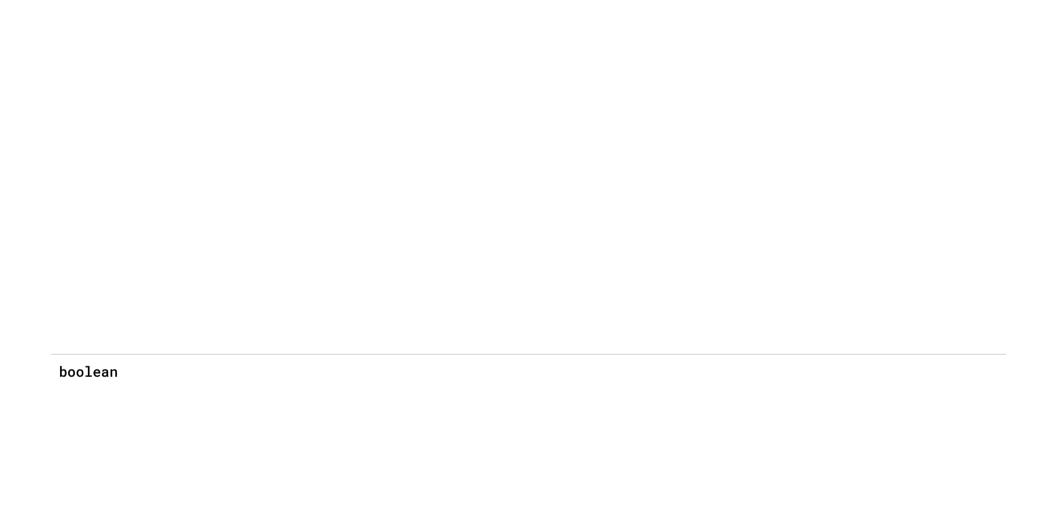




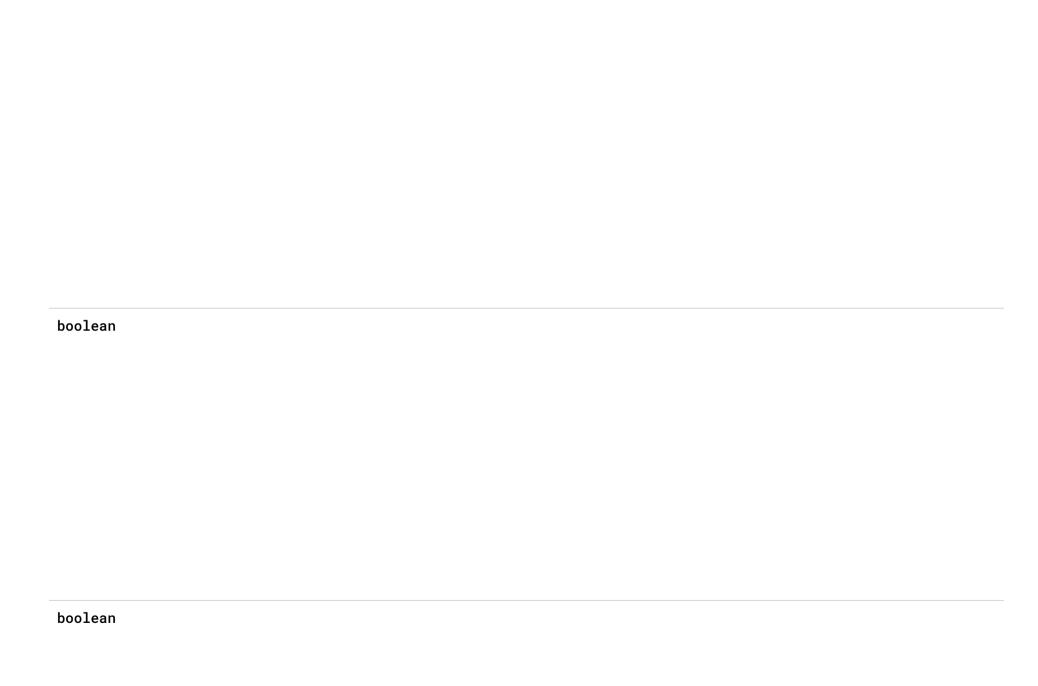


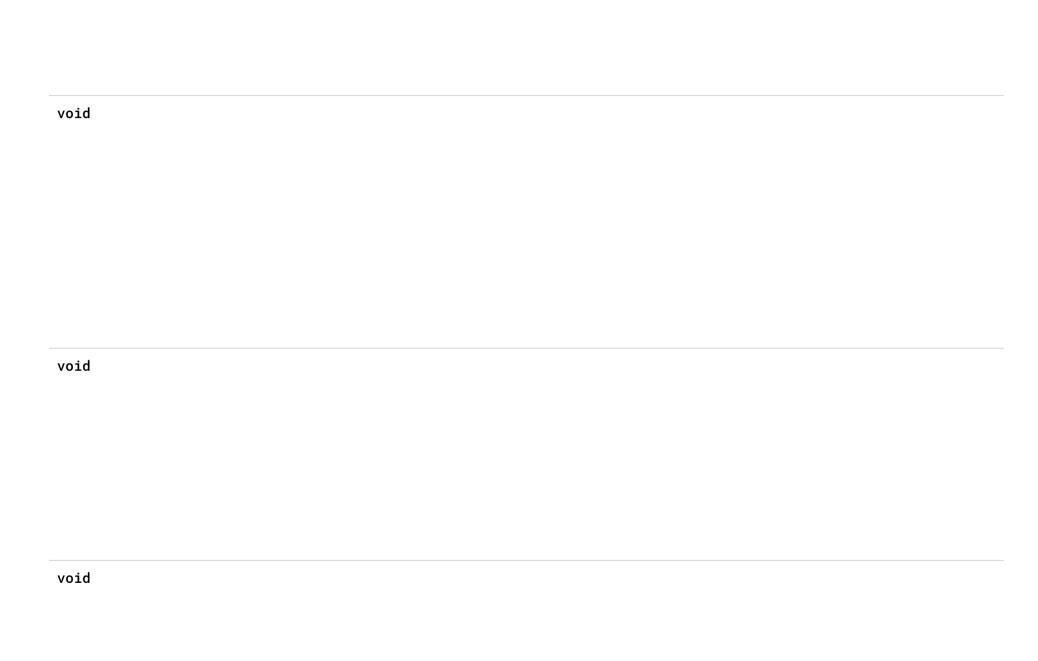
void

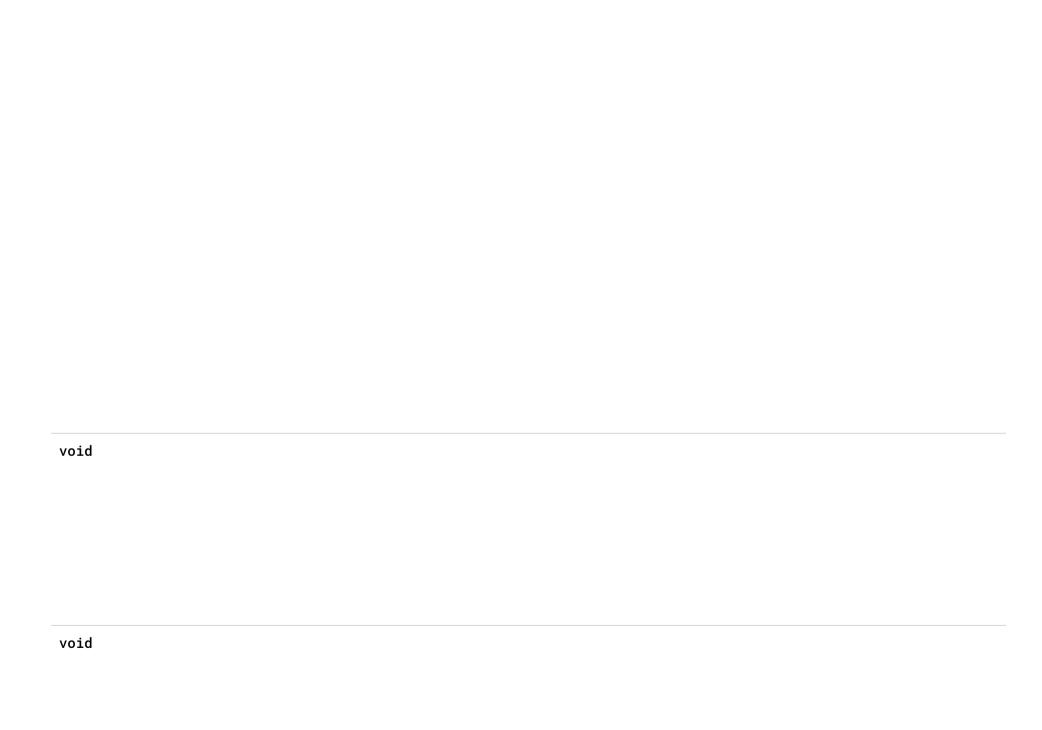




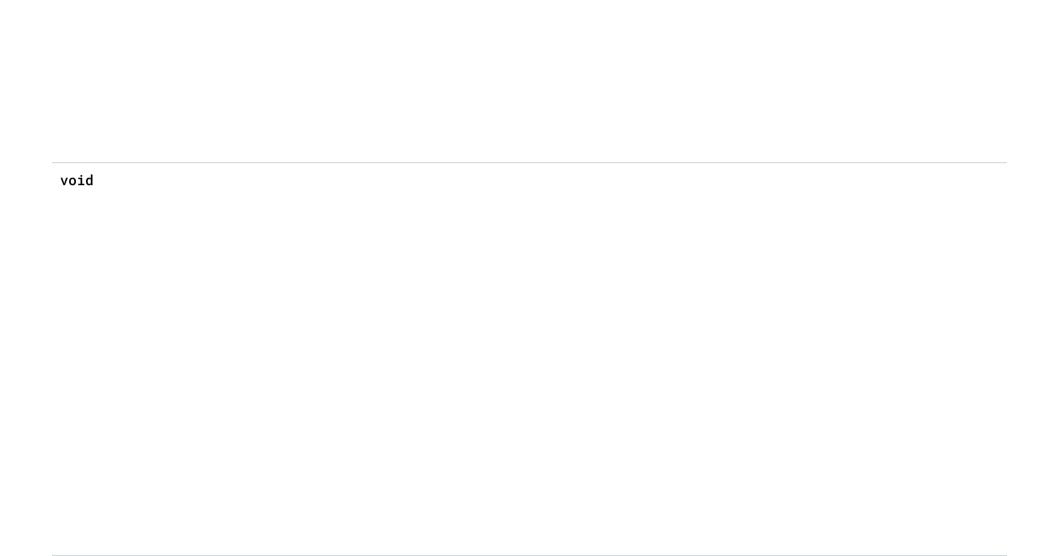
boolean



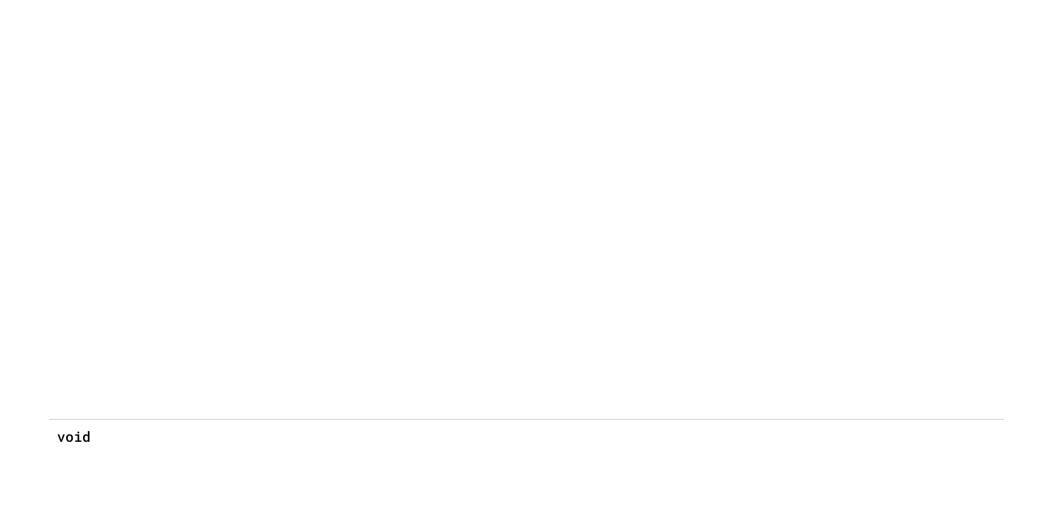


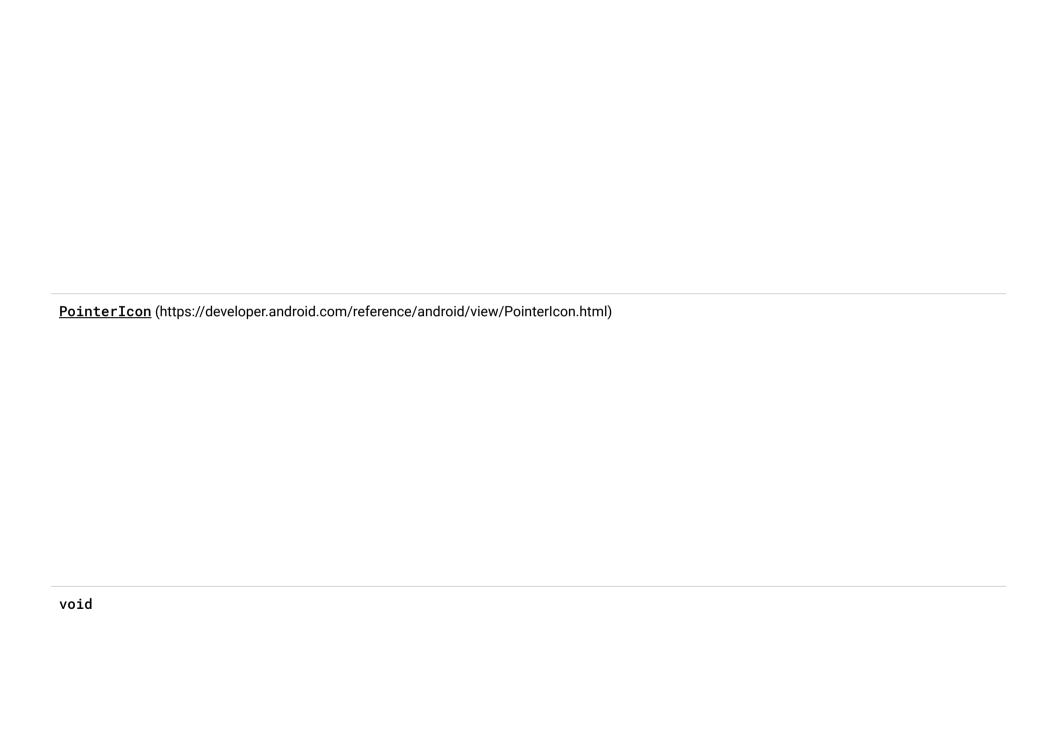


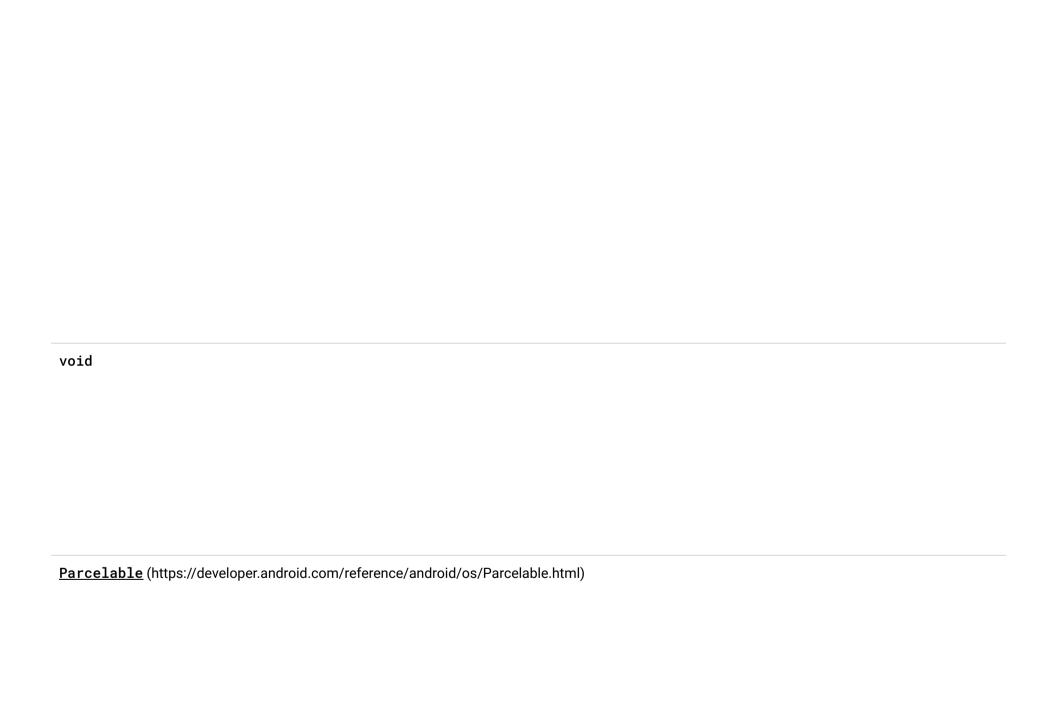


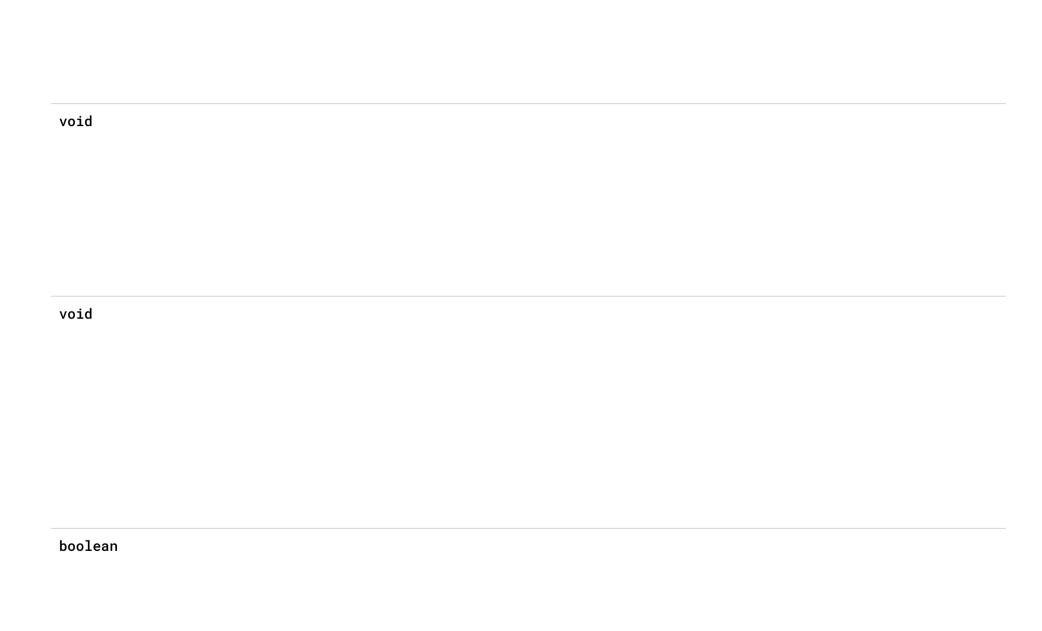


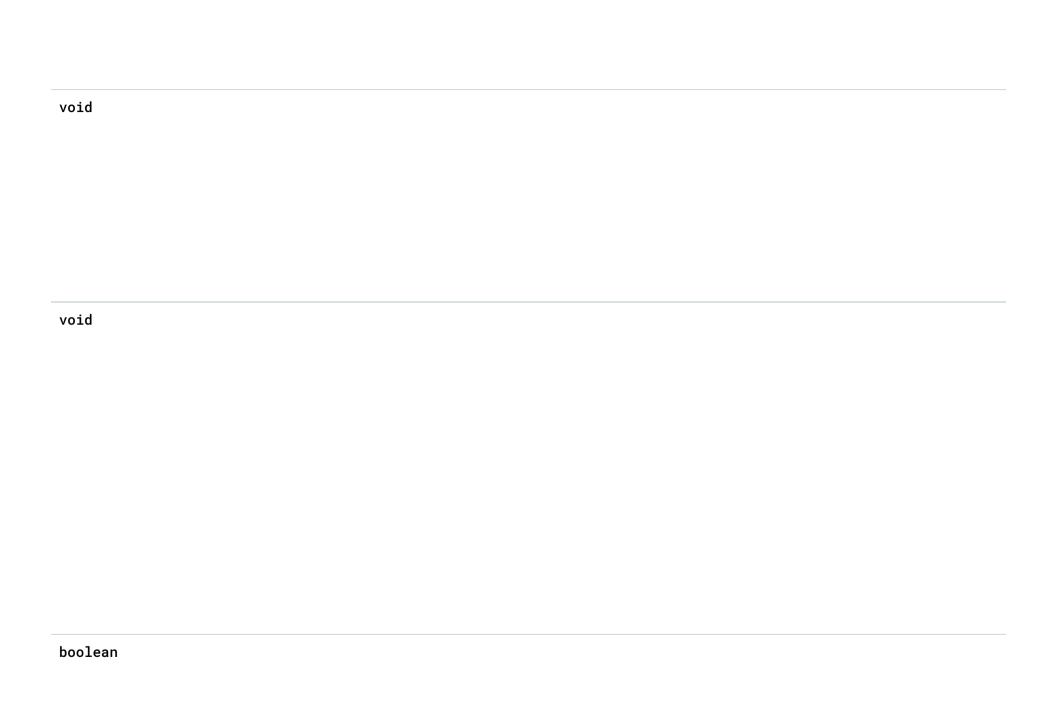
void

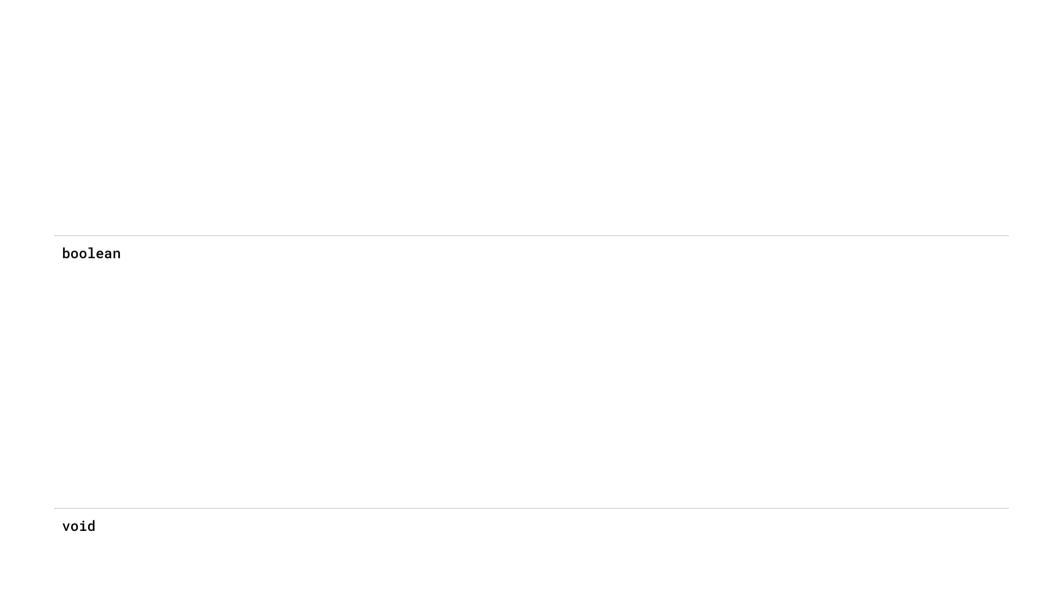


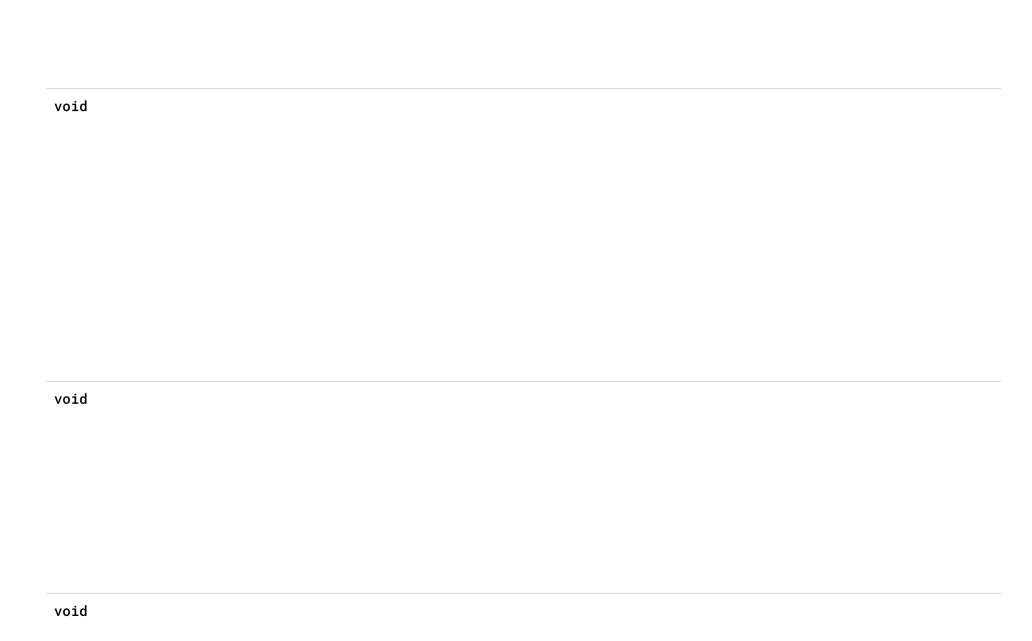


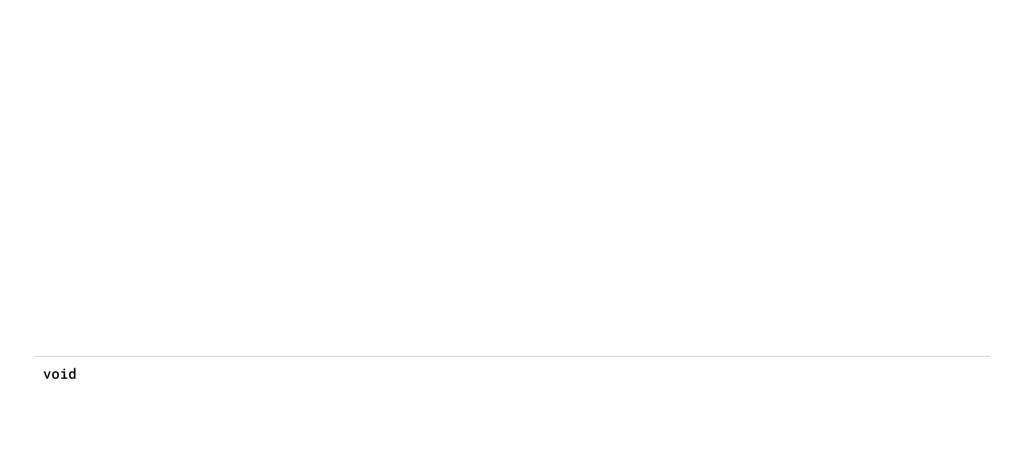


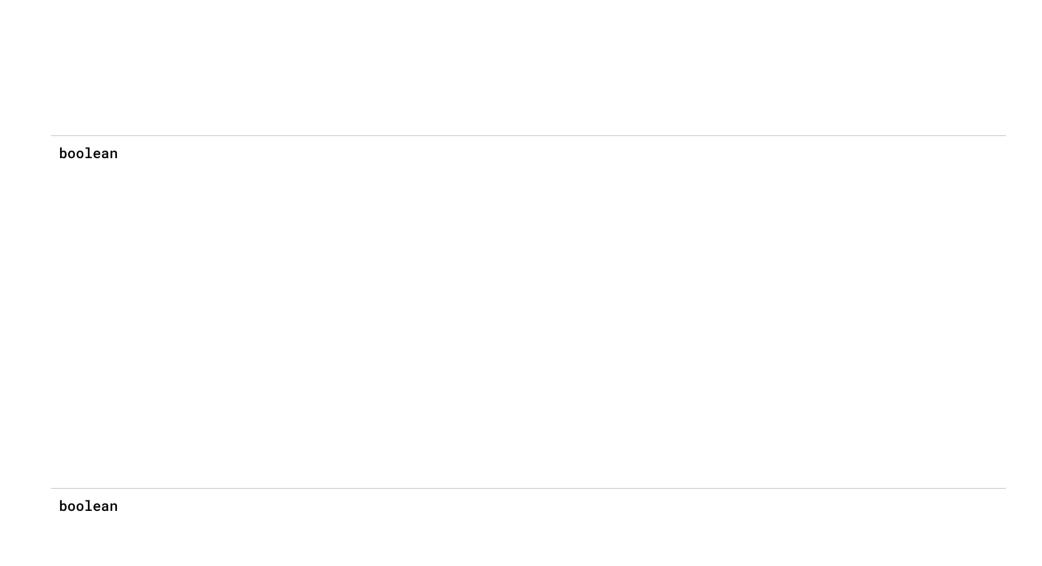


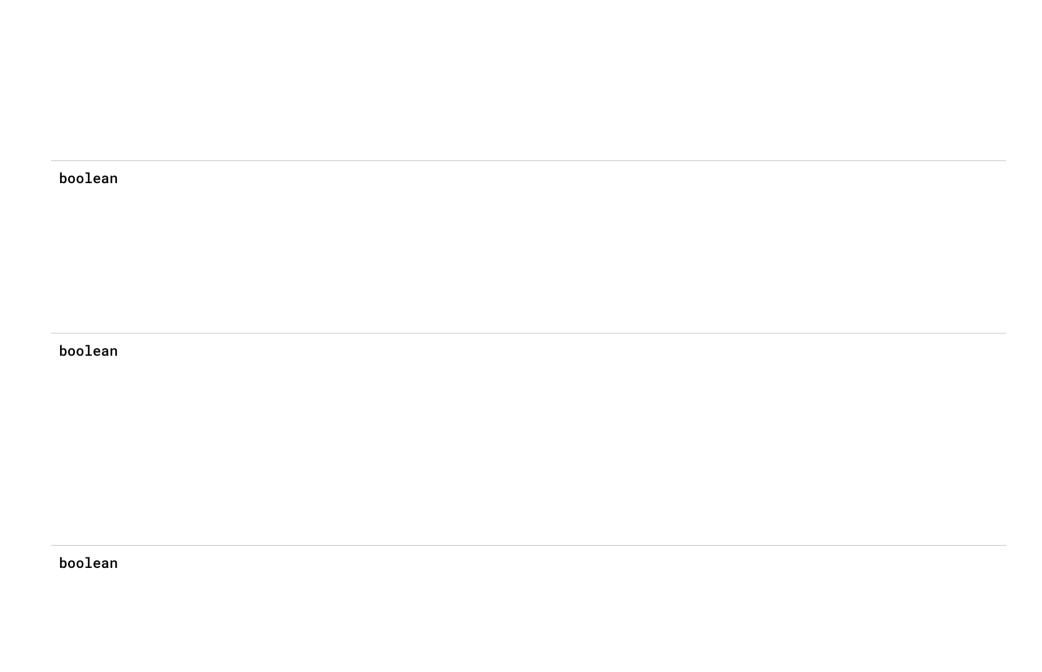


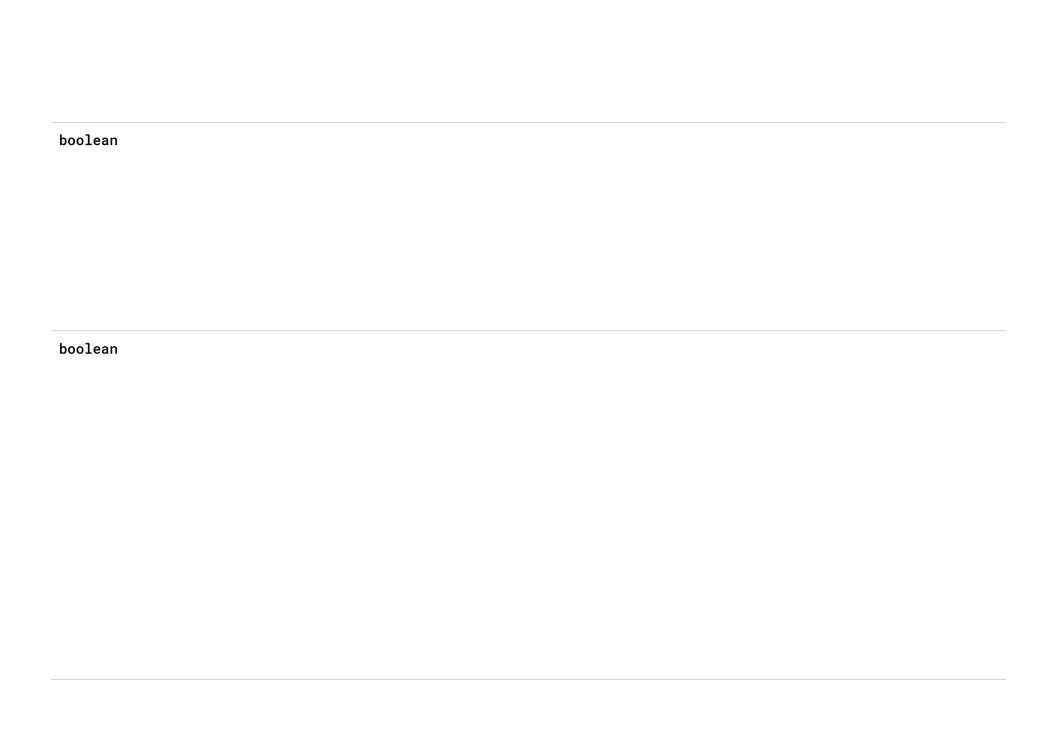




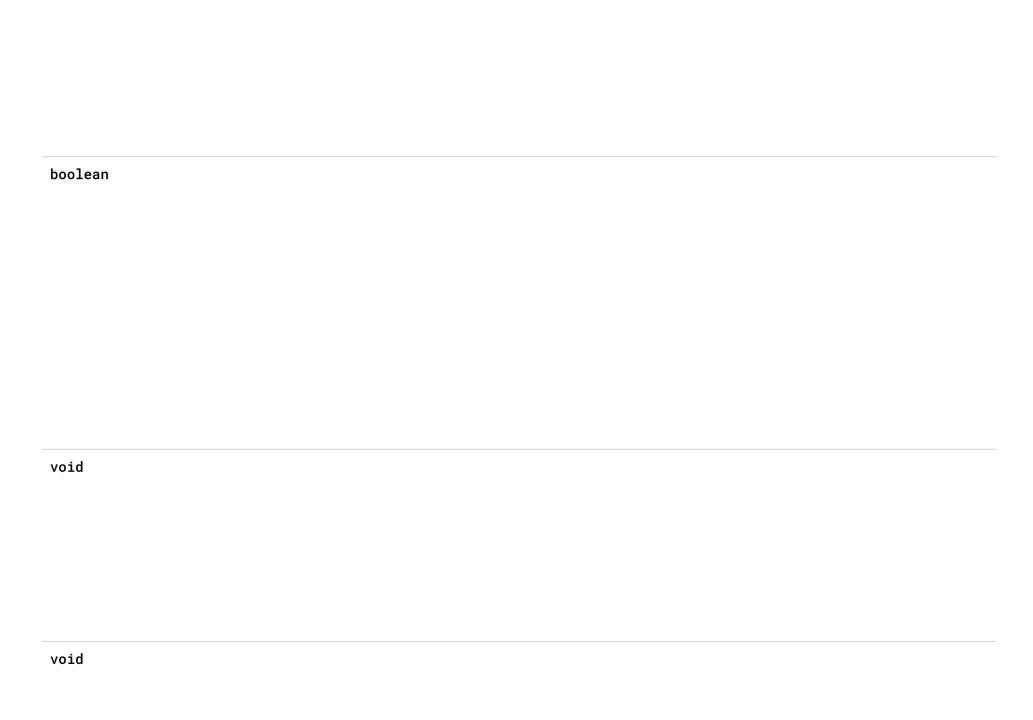


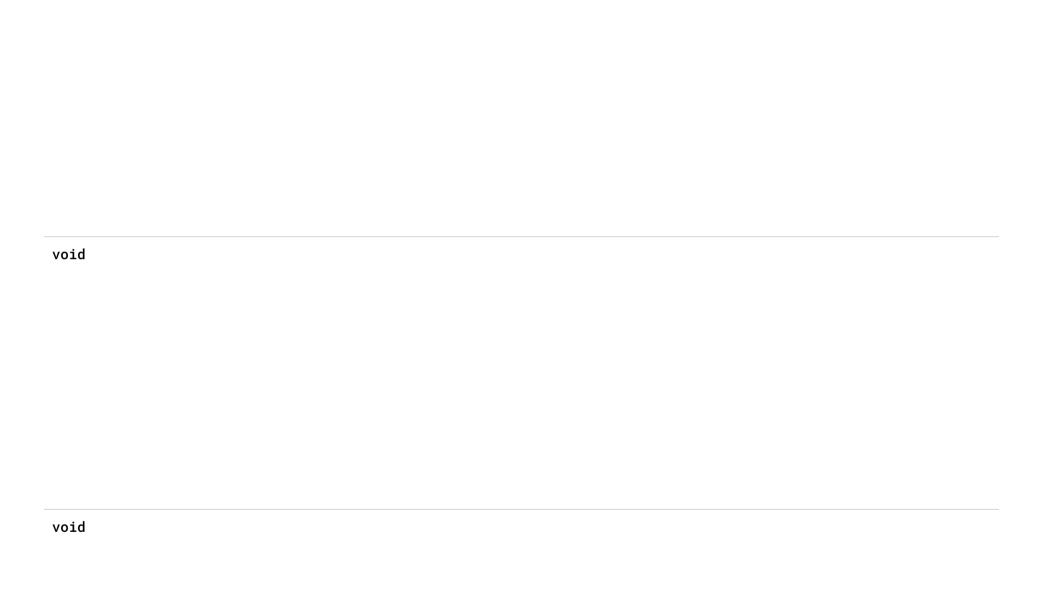


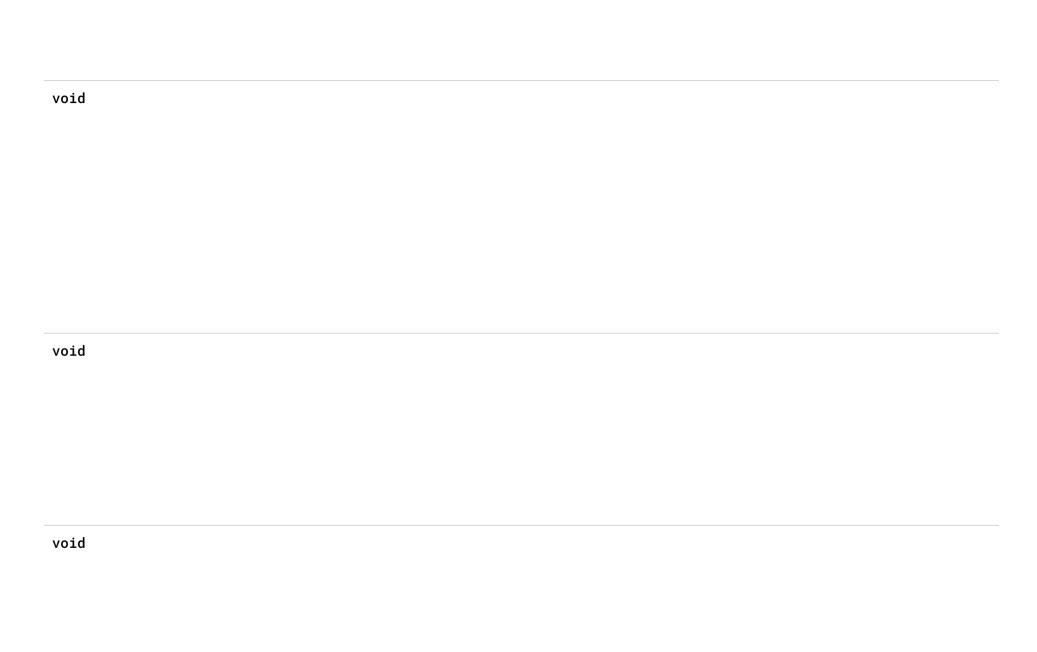


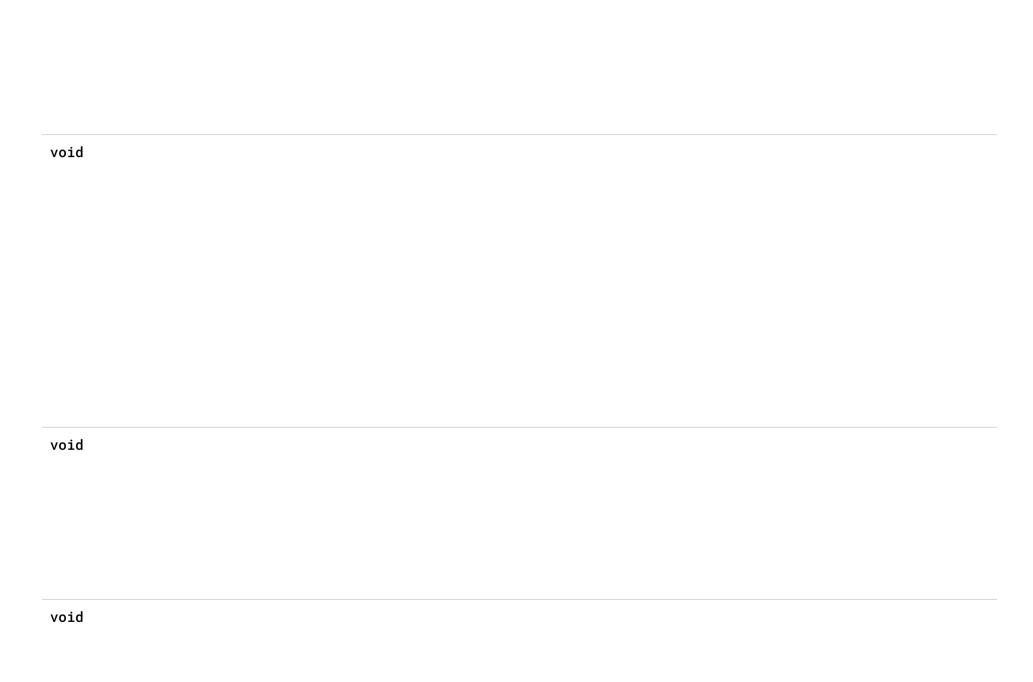


boolean		
boolean		
void		
boolean		



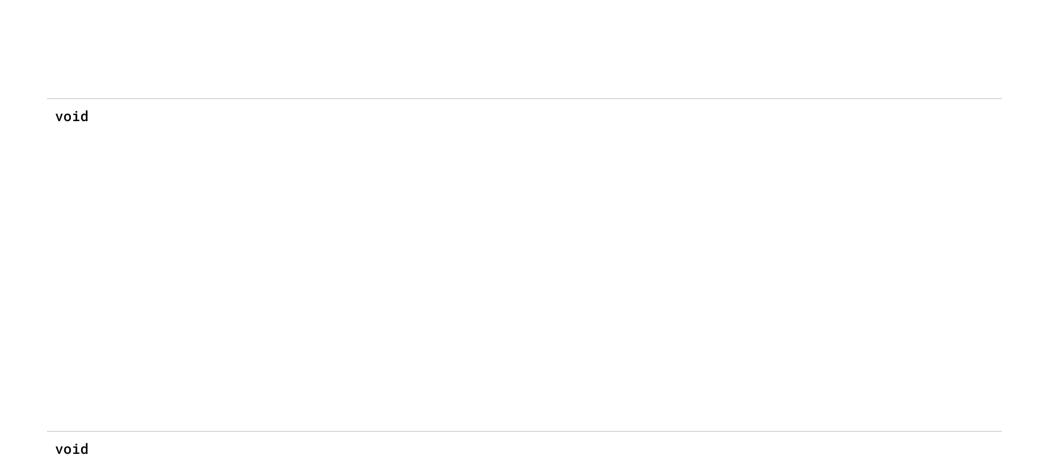


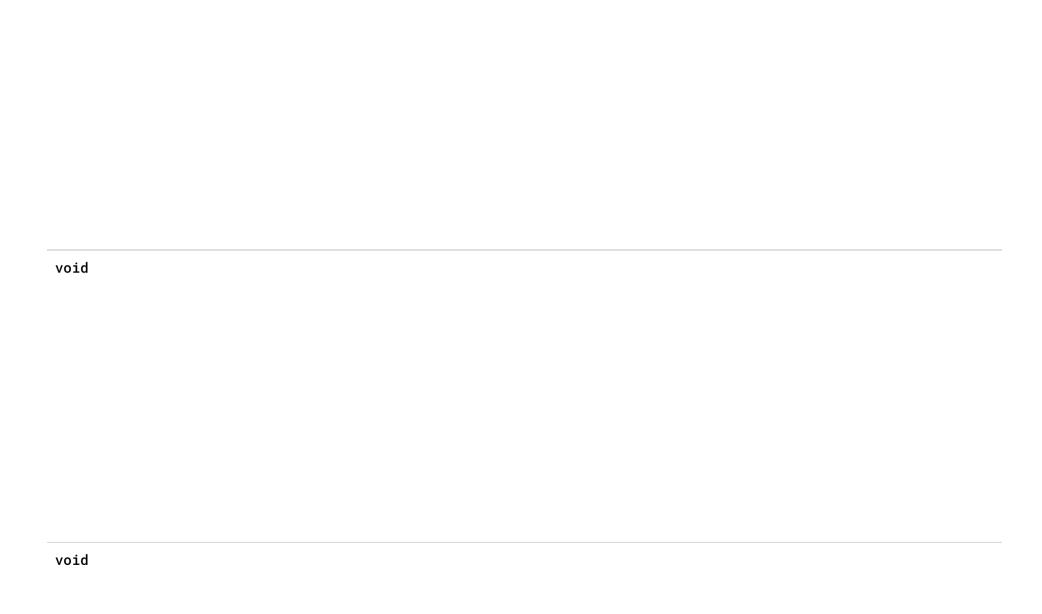


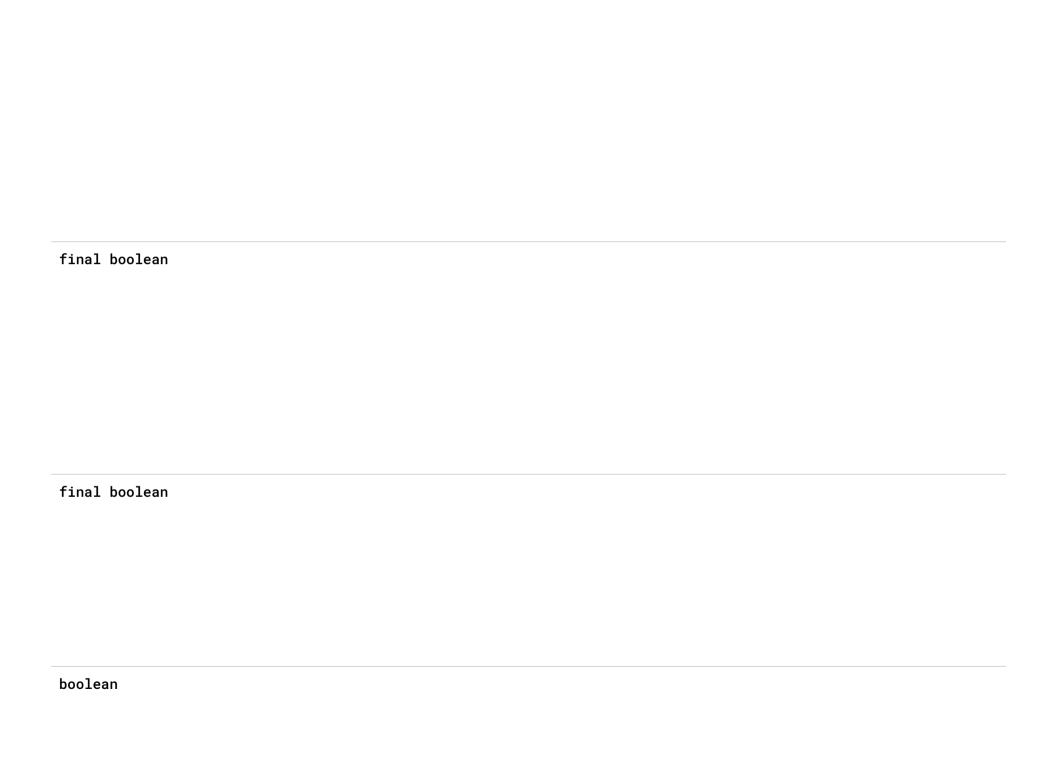


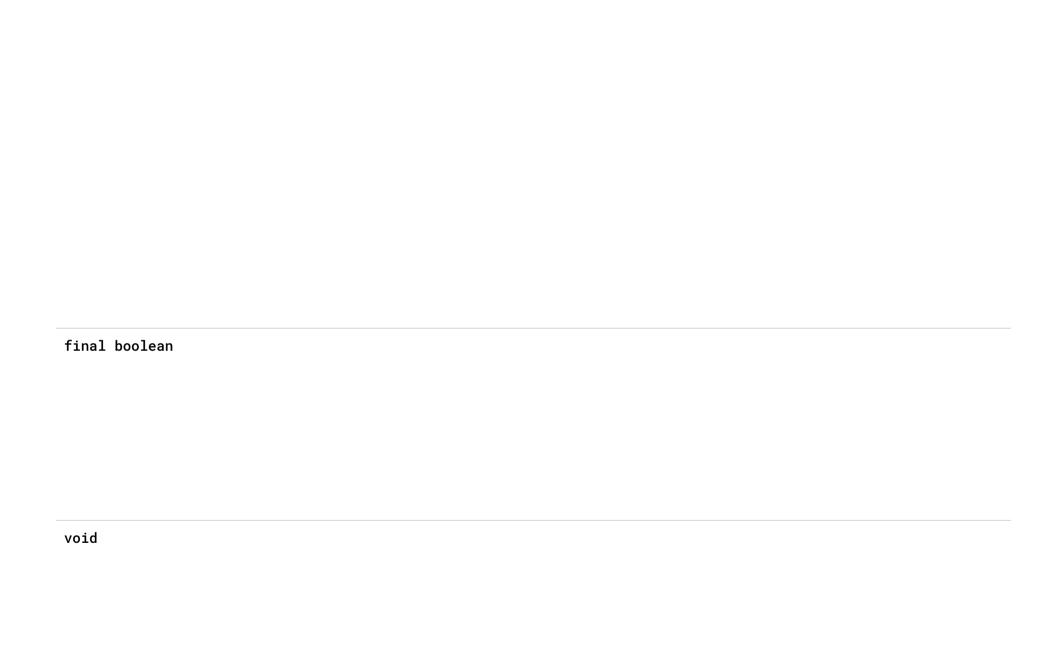
boolean

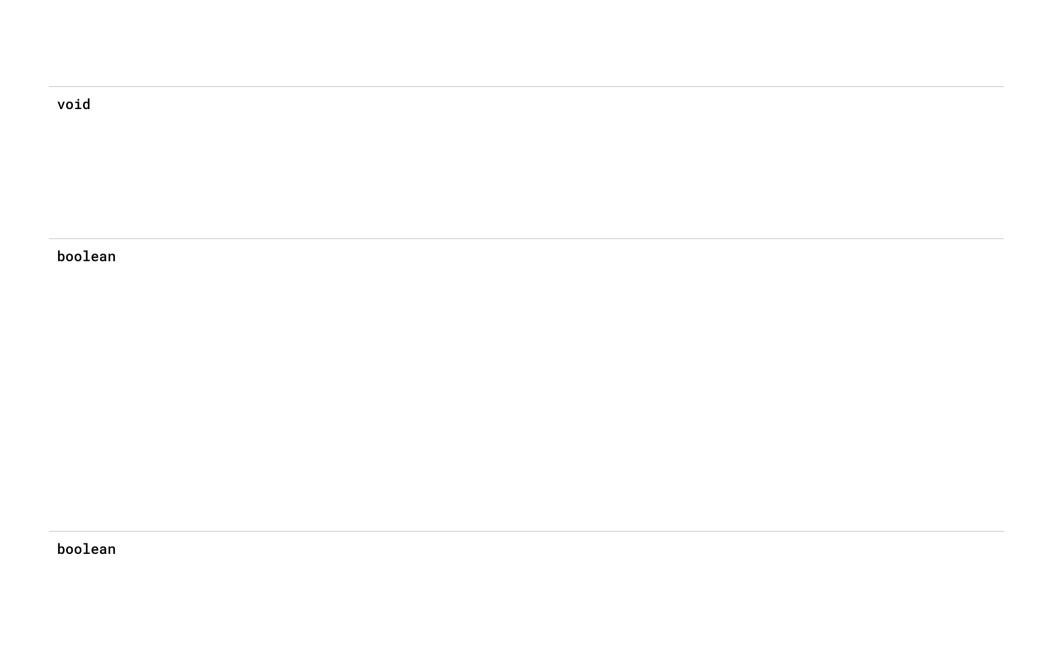
void

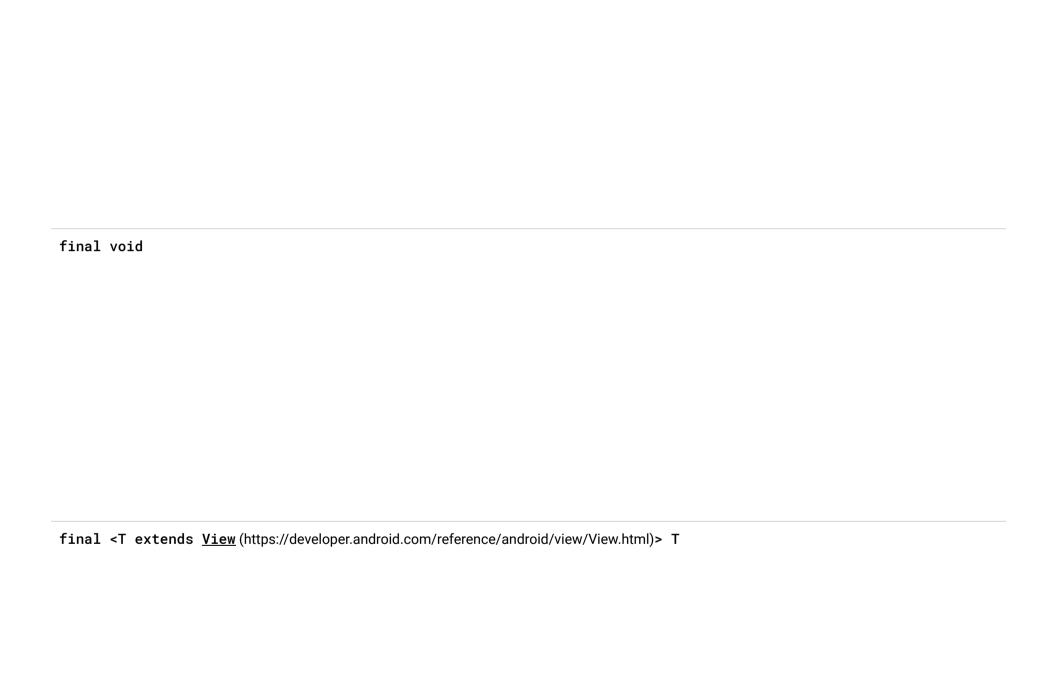












void

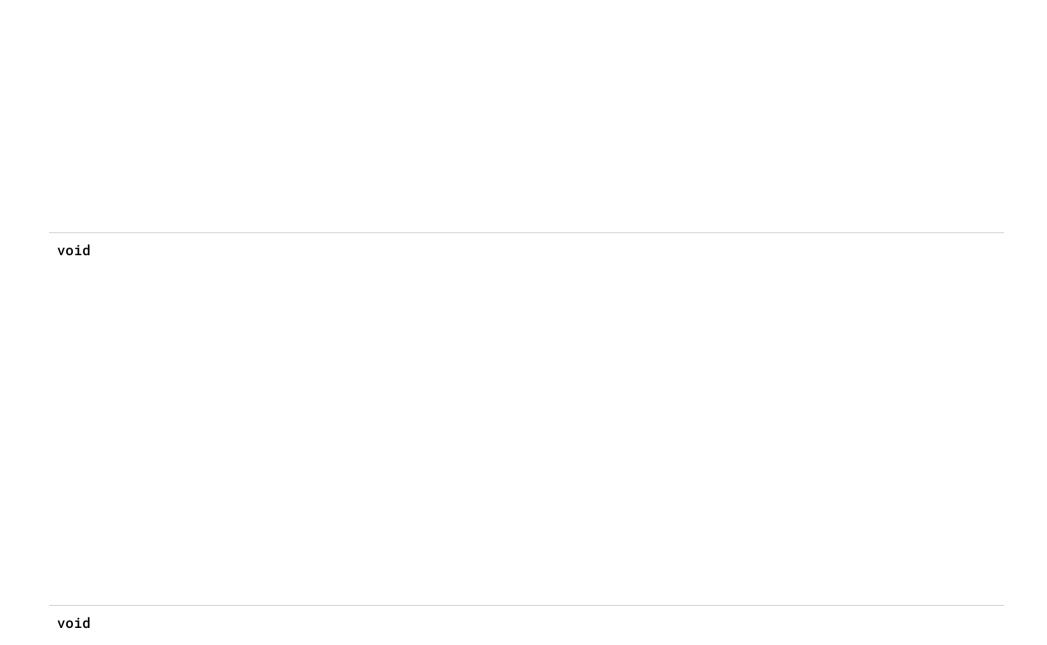
static int

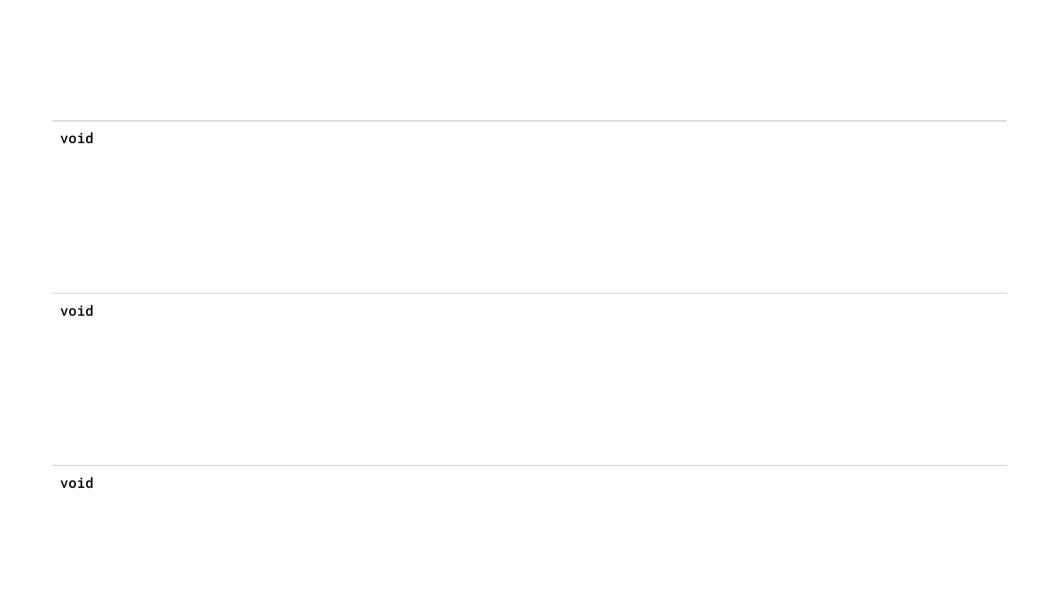
static int

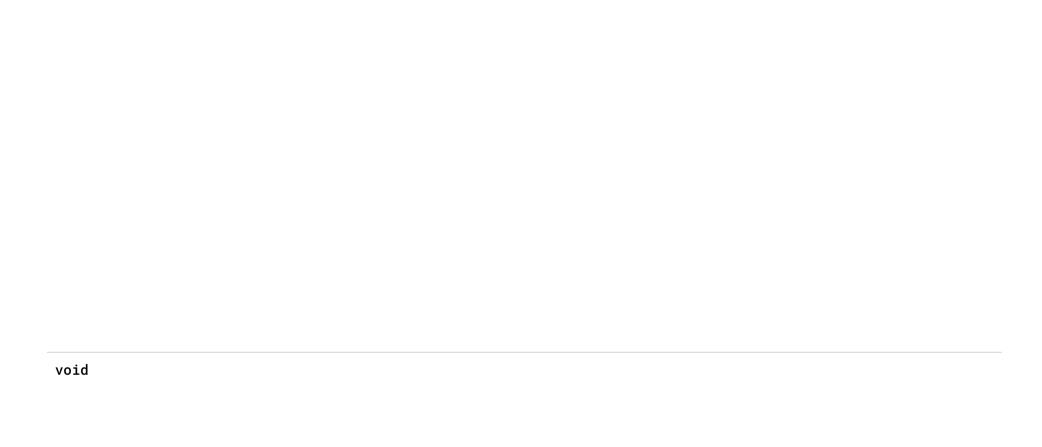
boolean

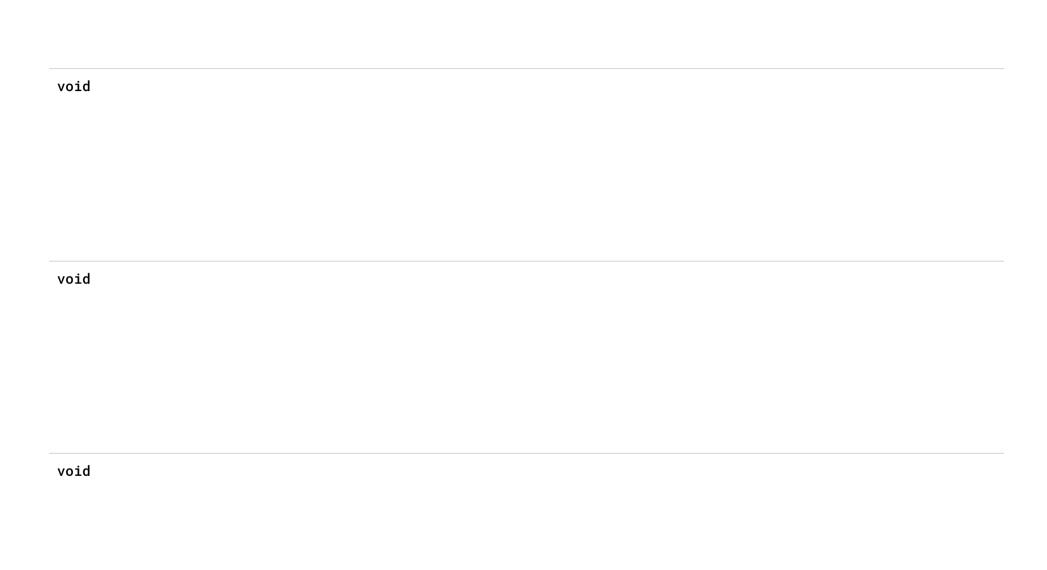
void

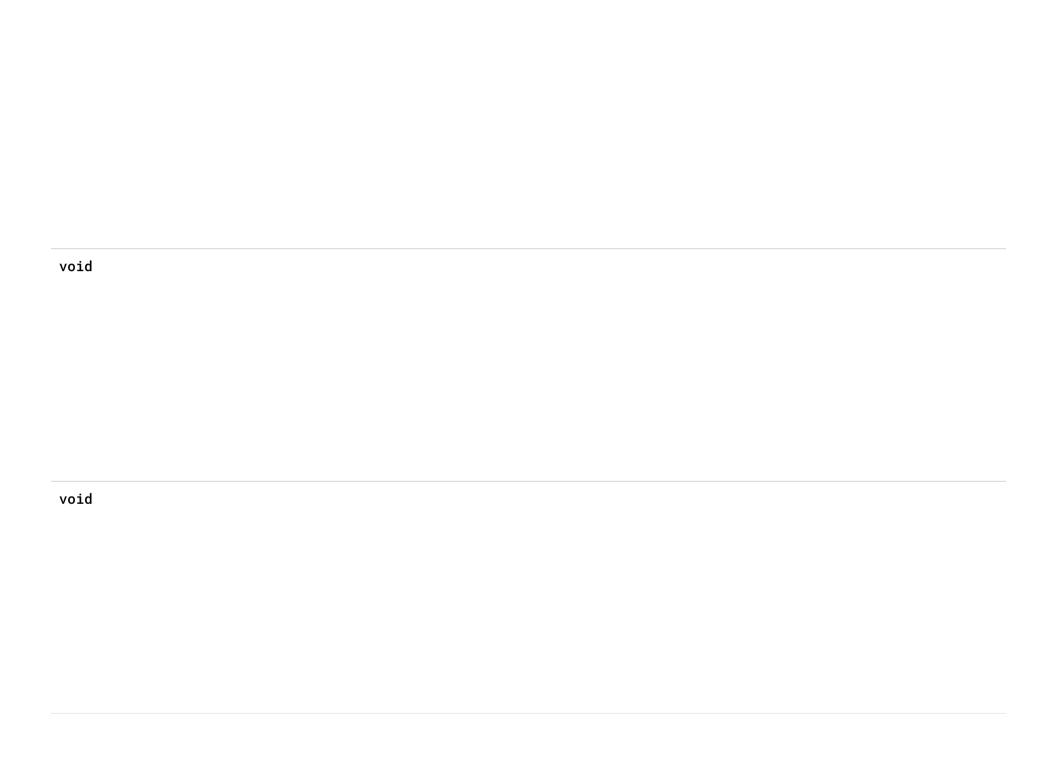
void

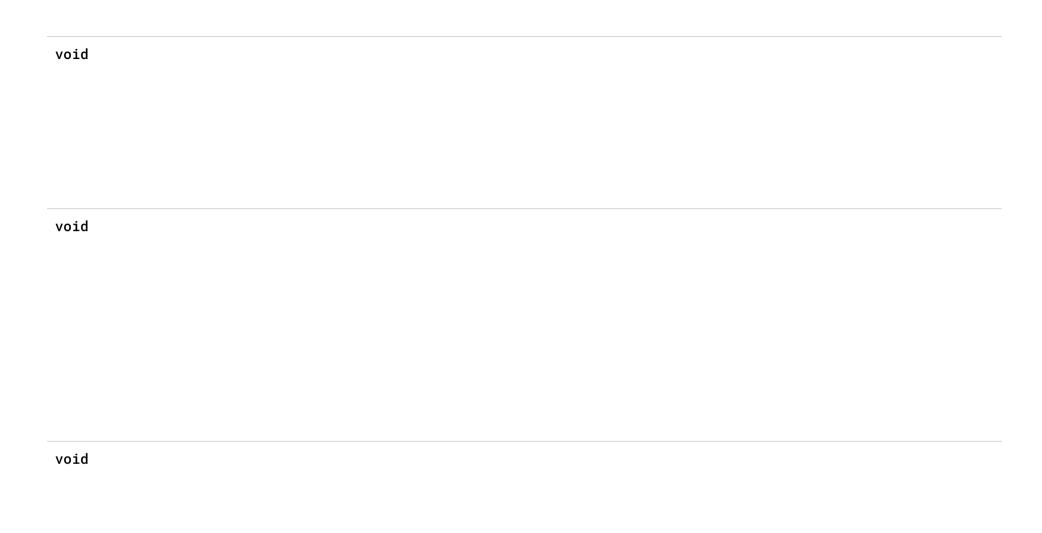


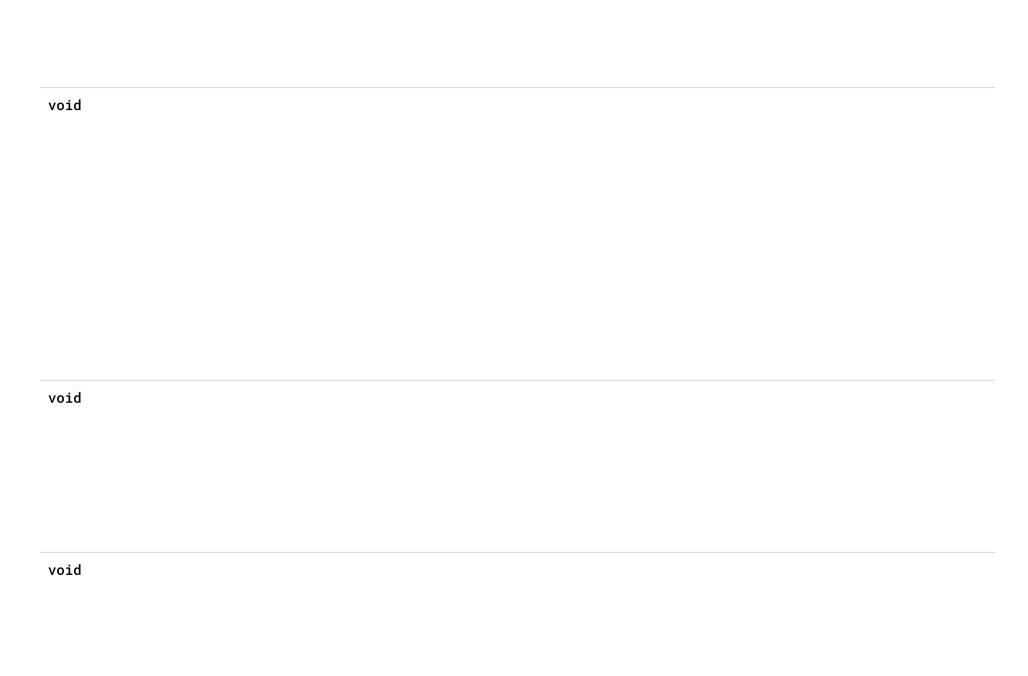


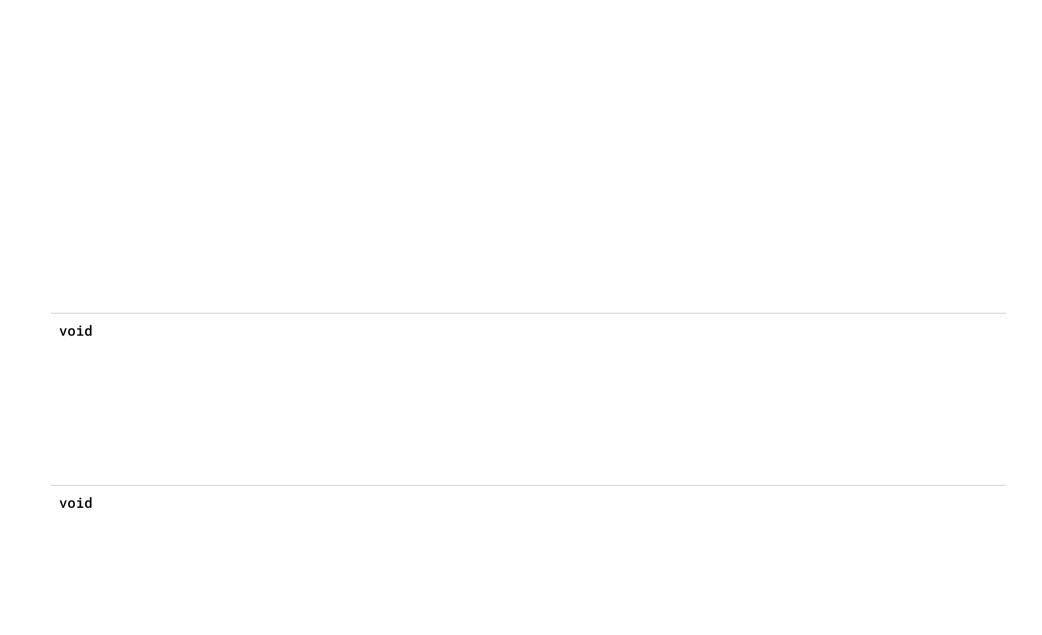




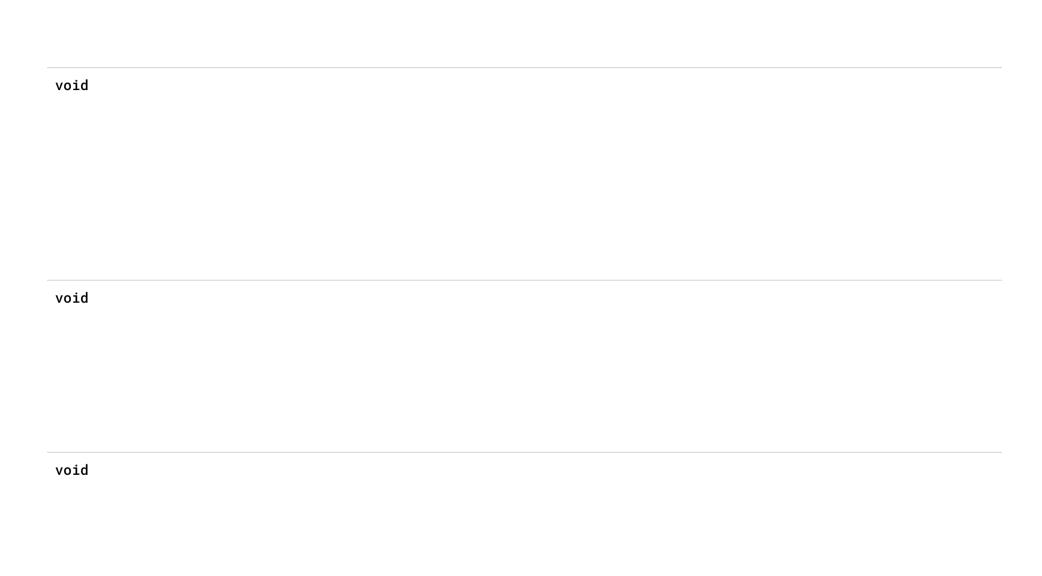


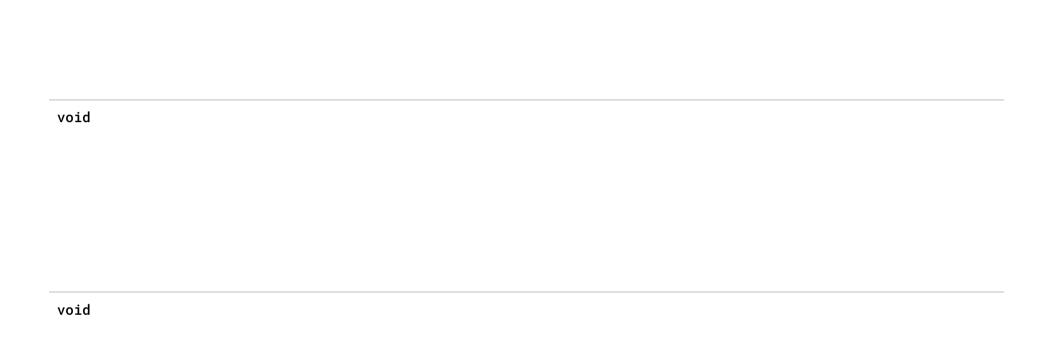


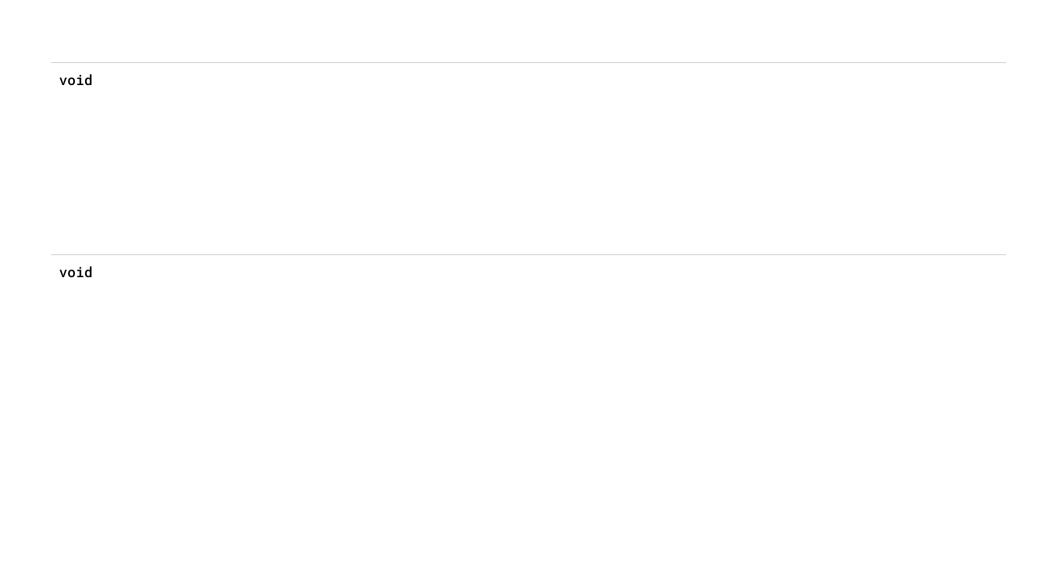


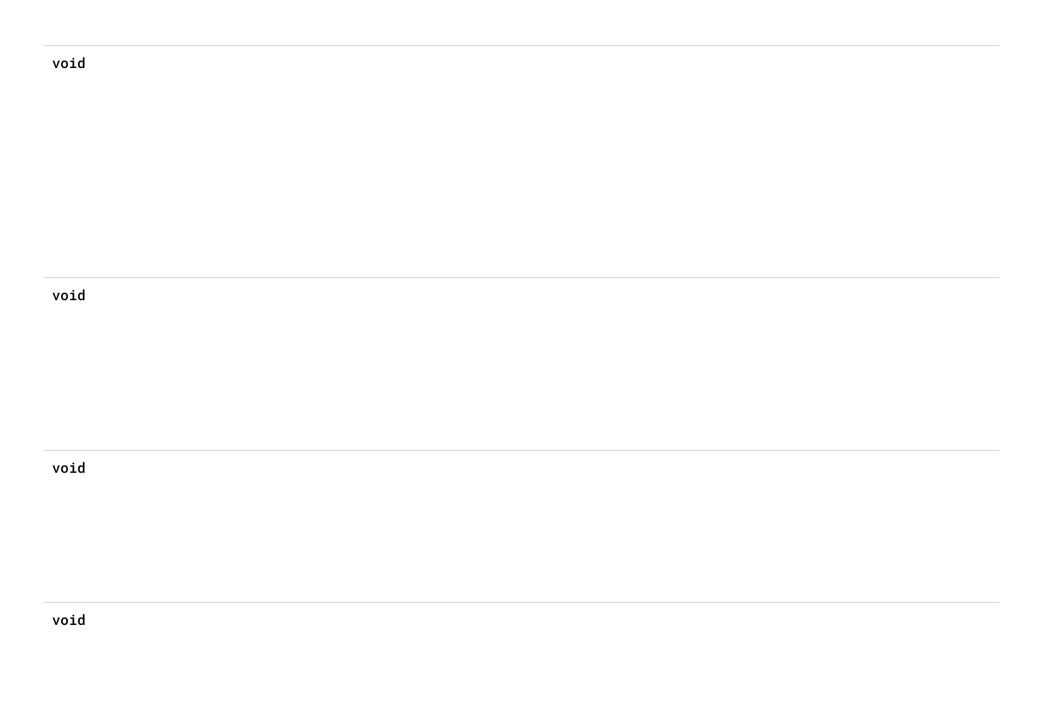


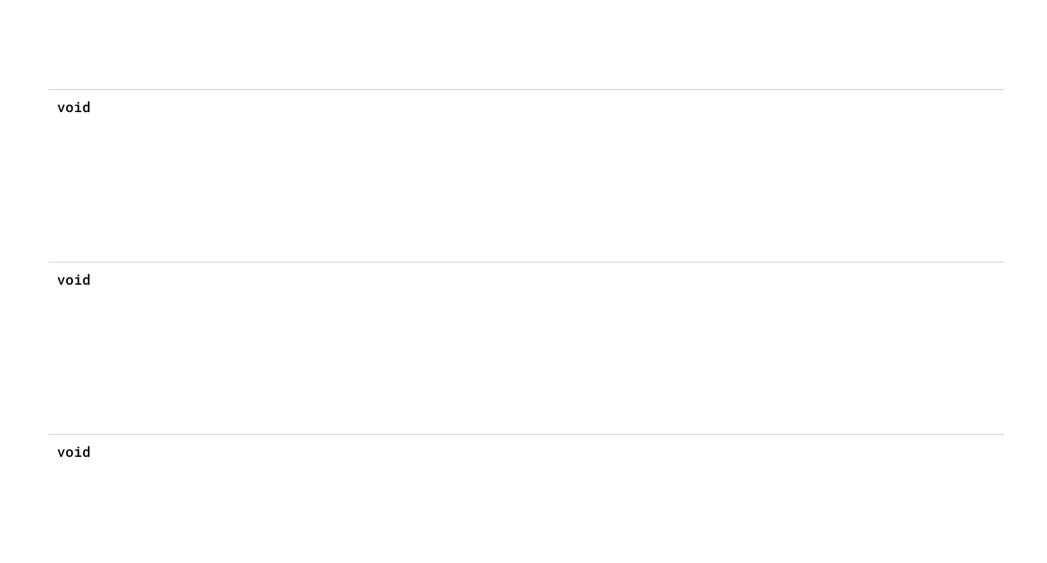
final void

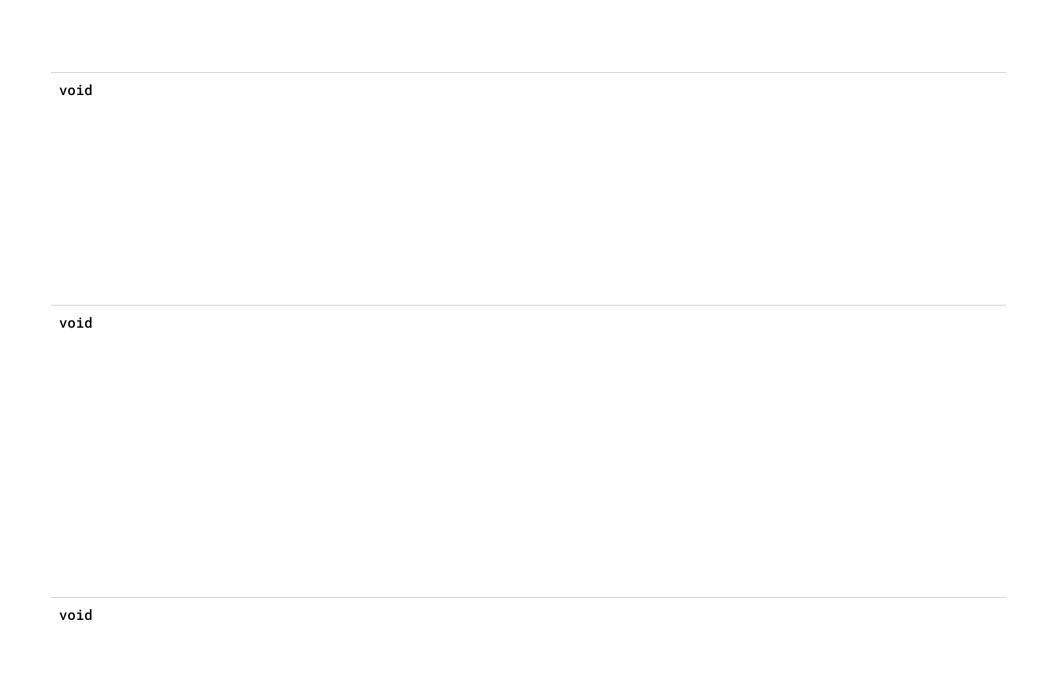


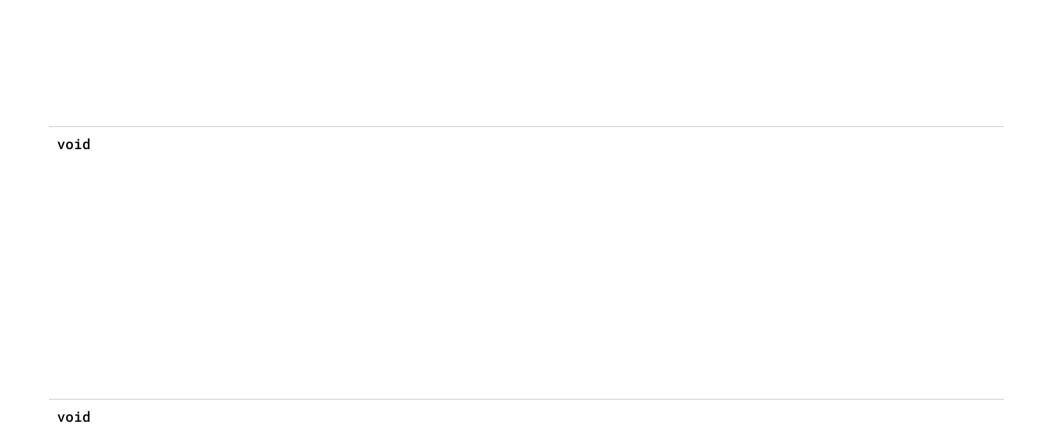


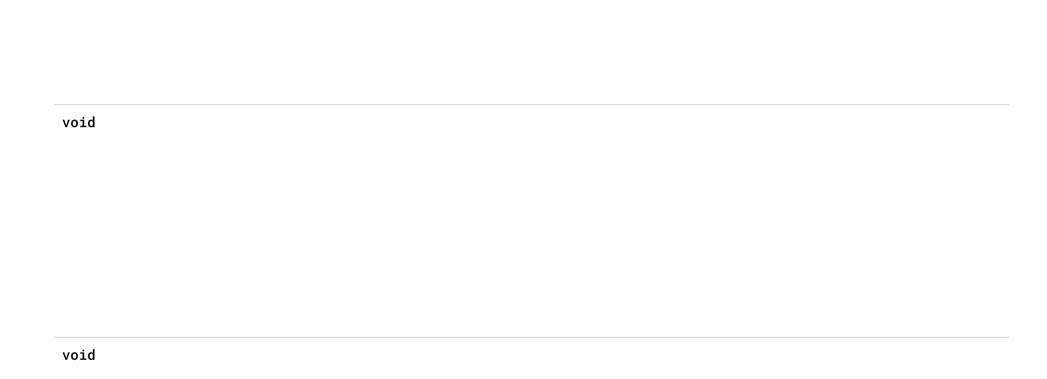


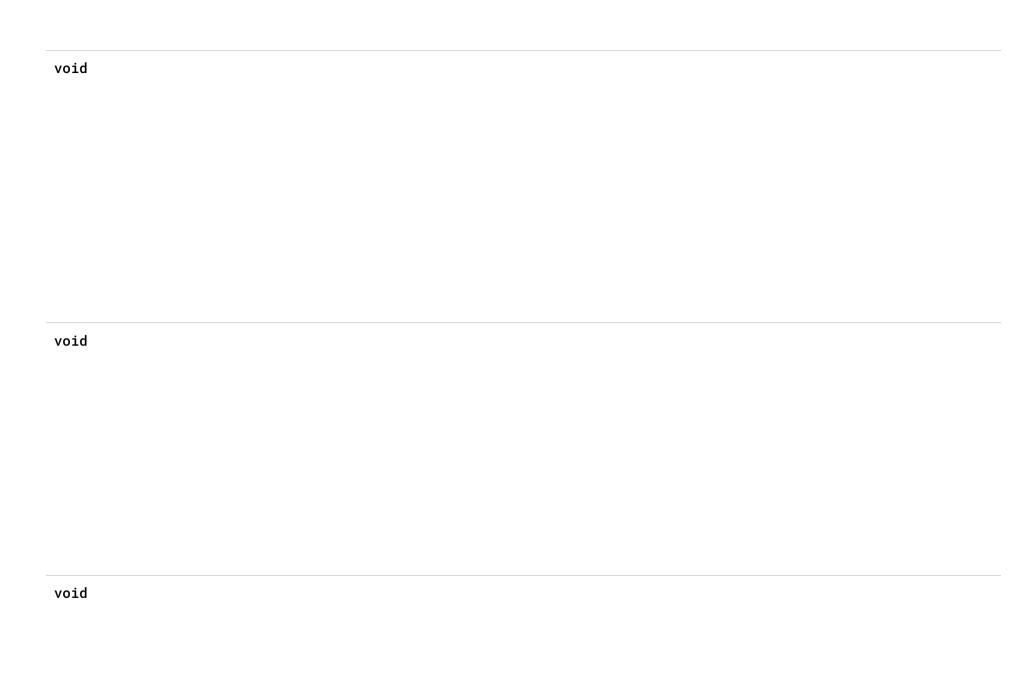


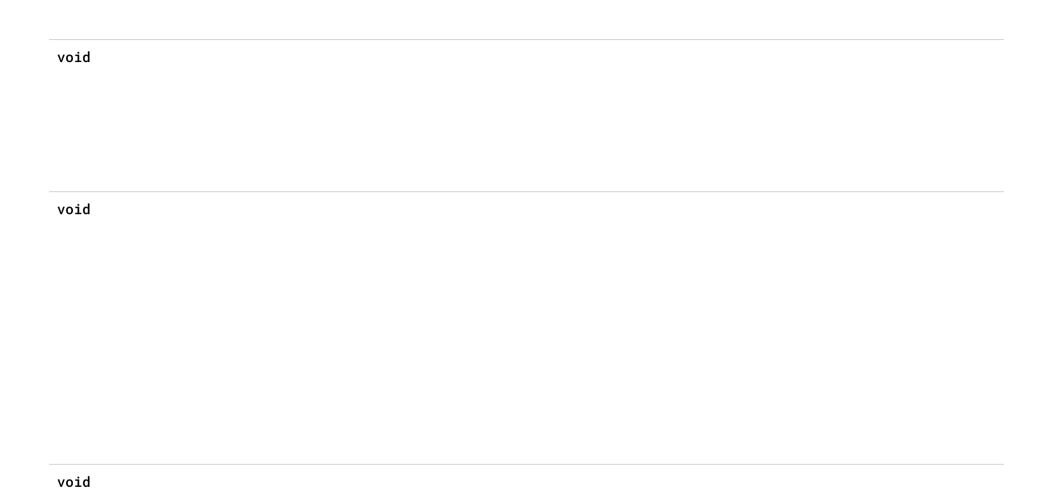


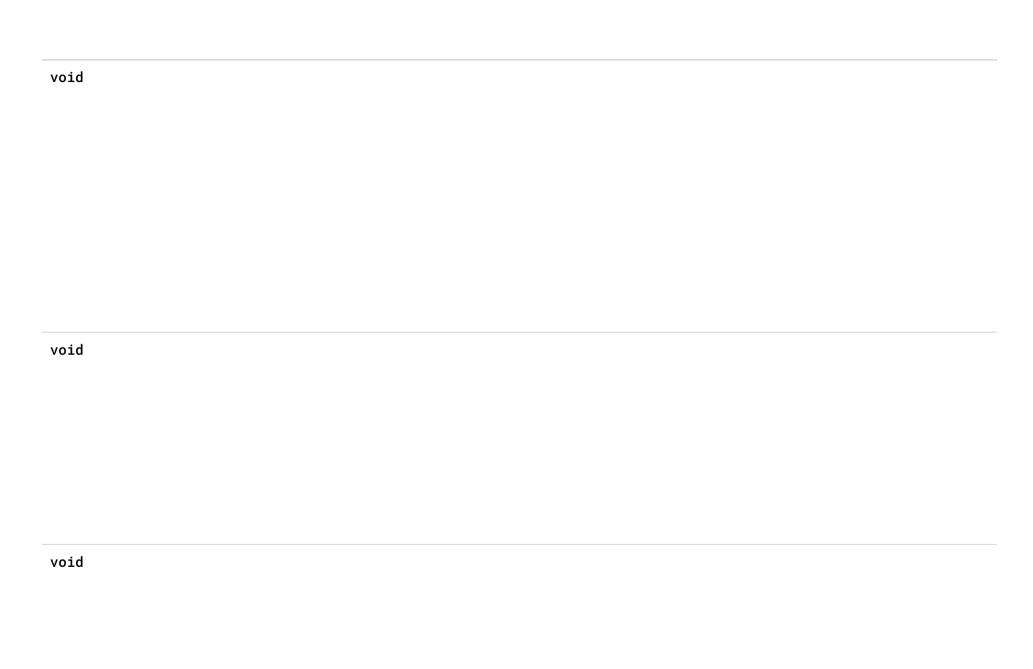


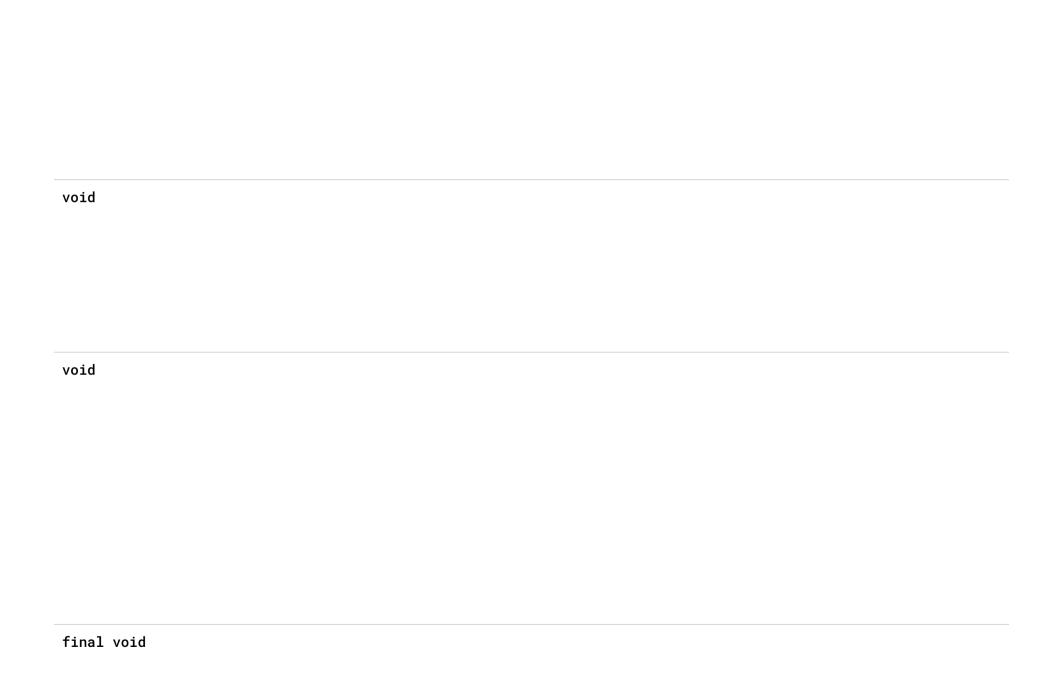


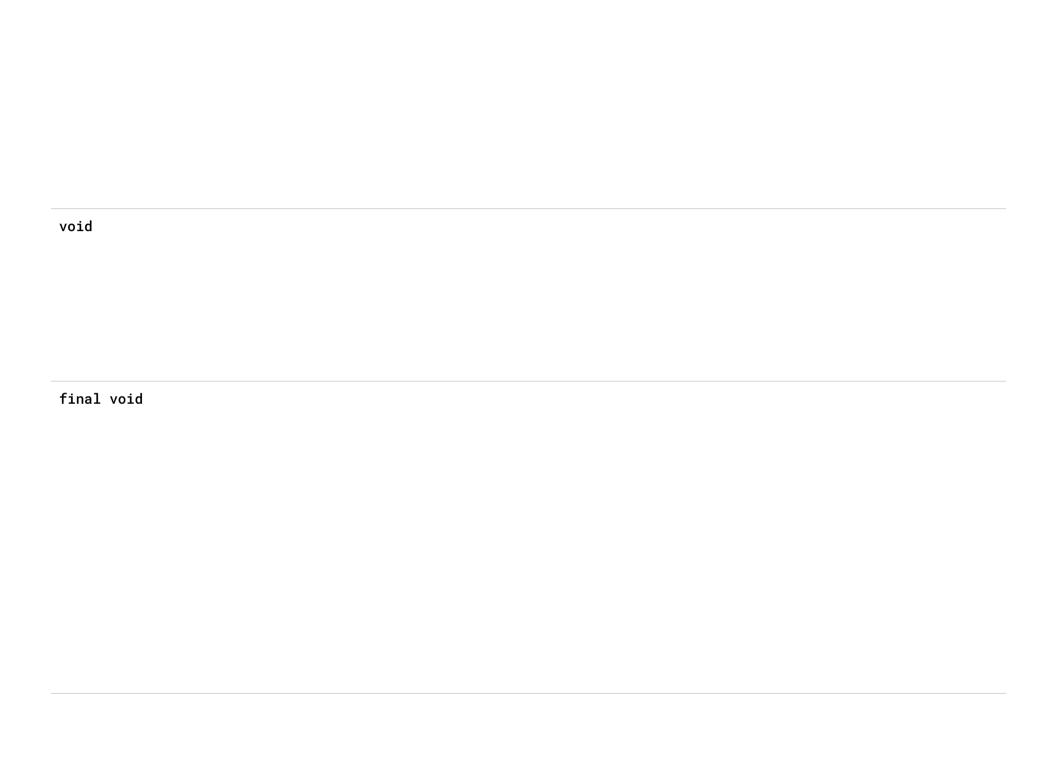


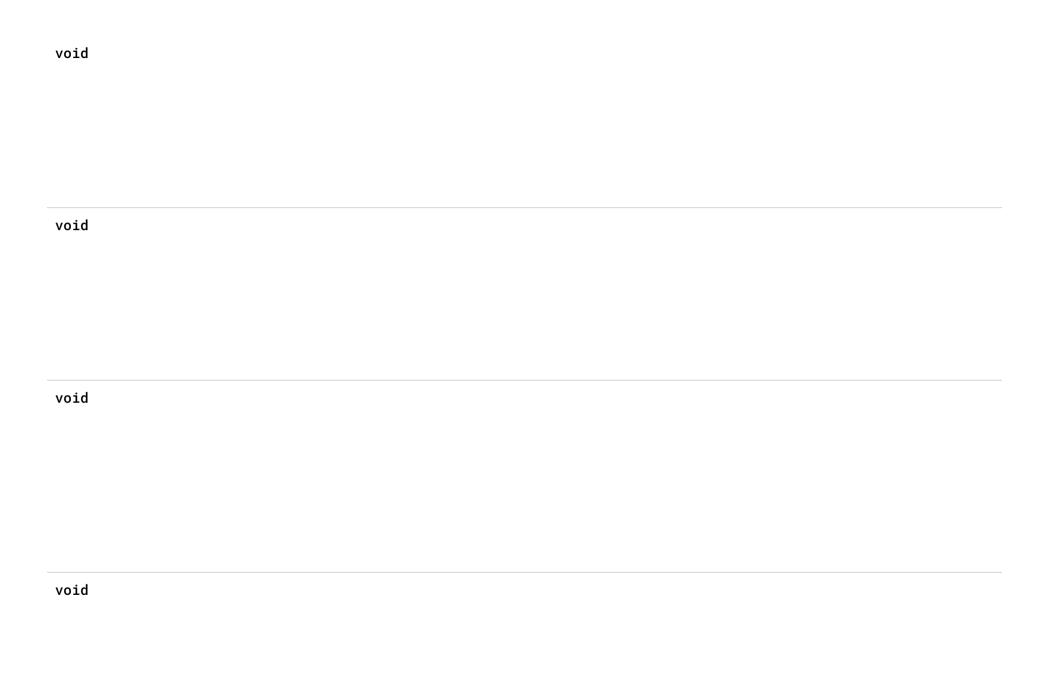


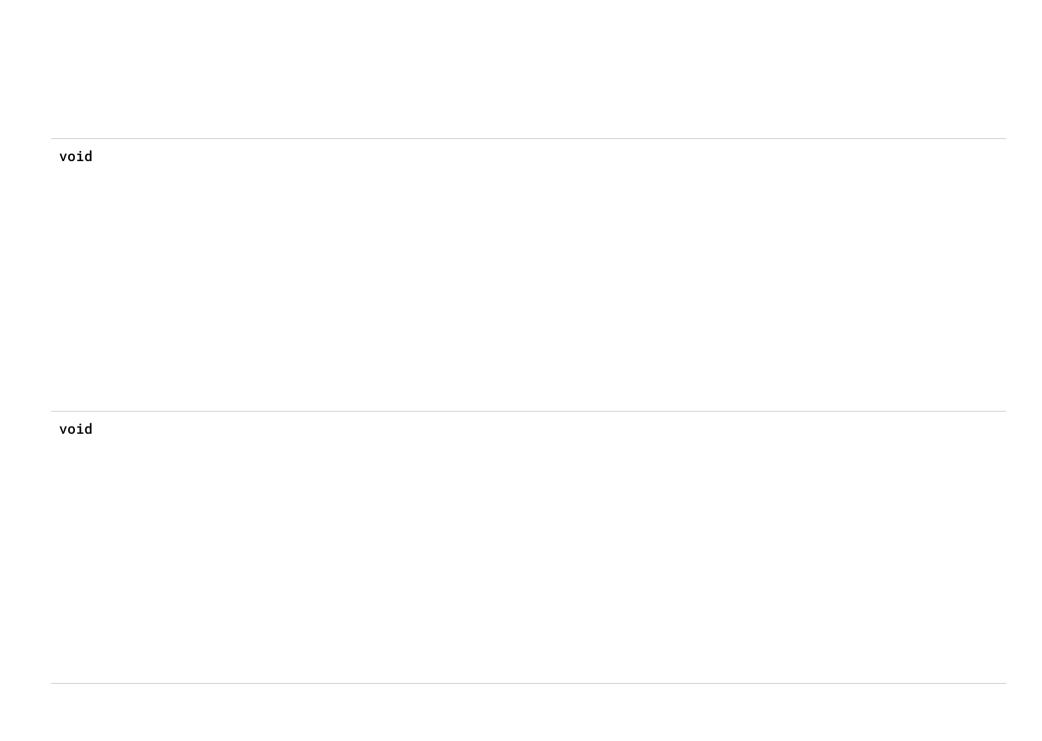


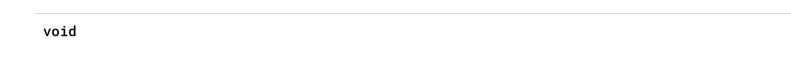


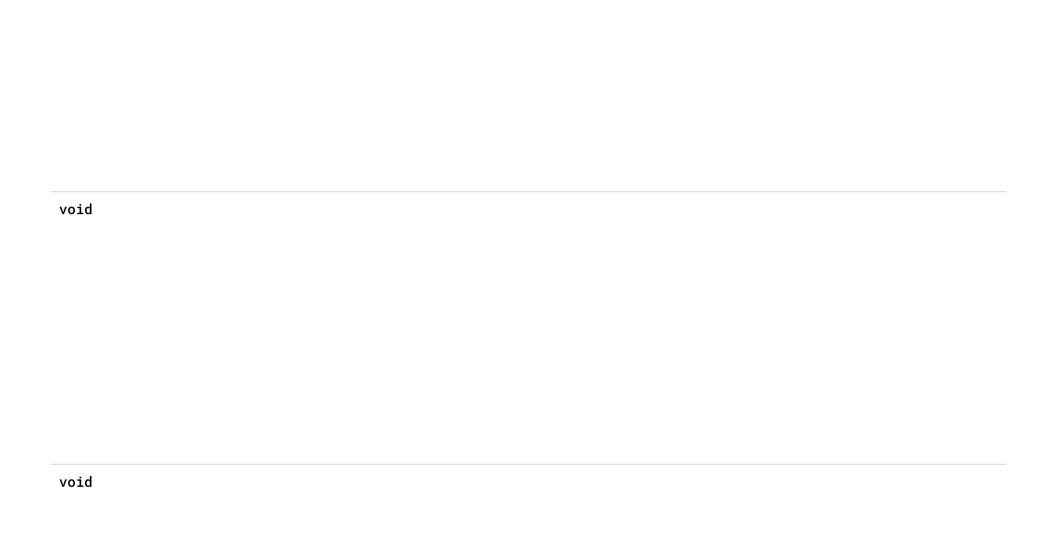




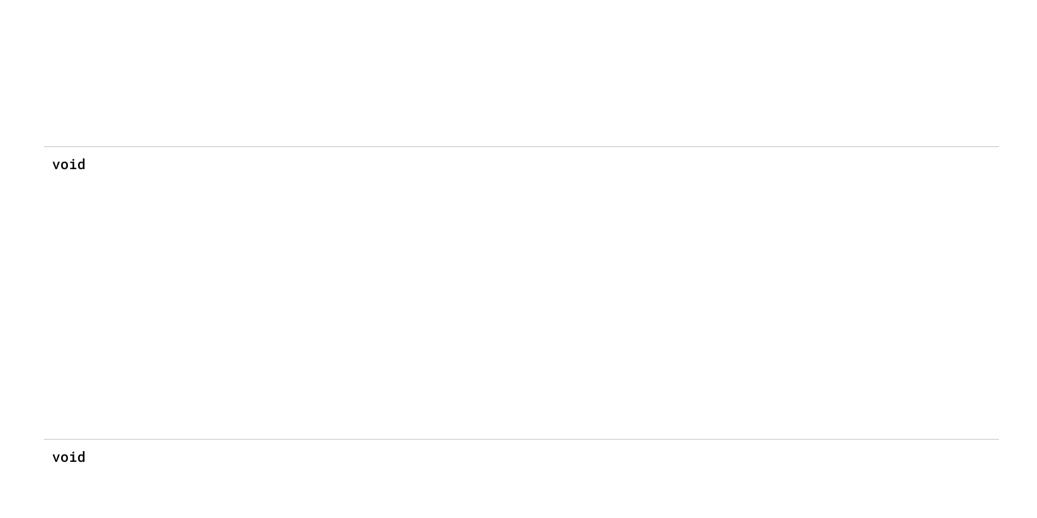


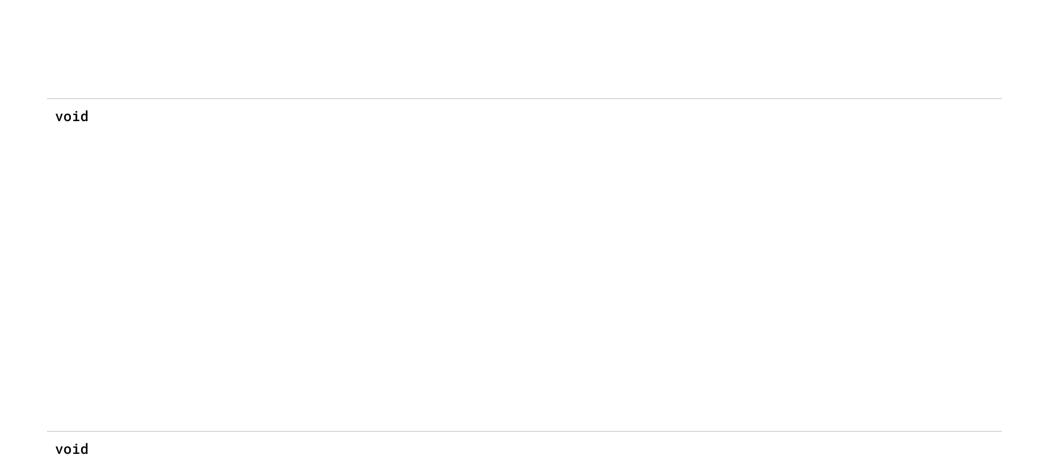


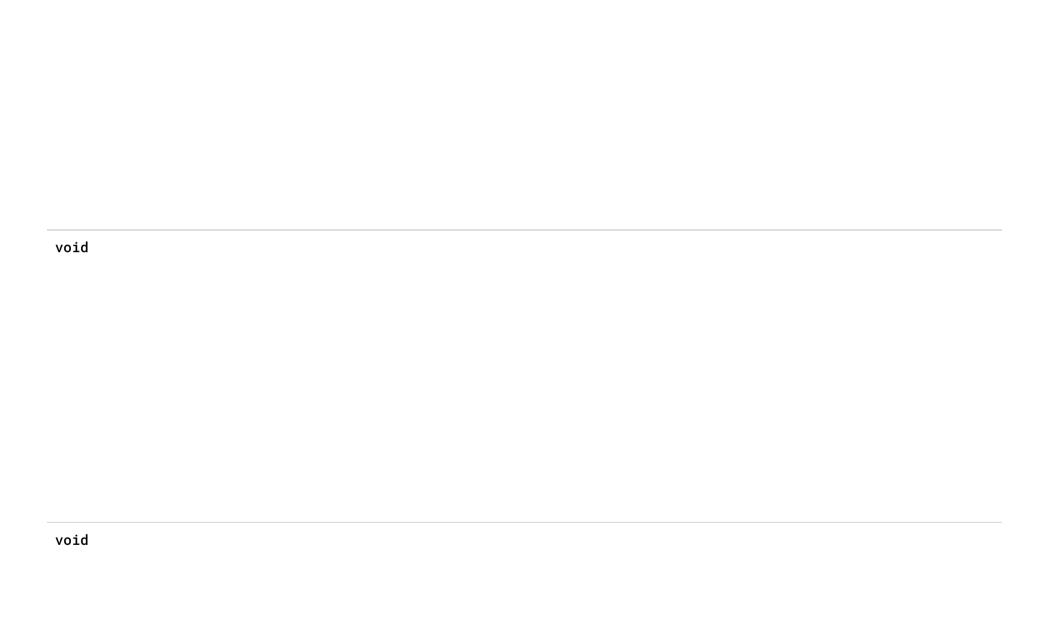


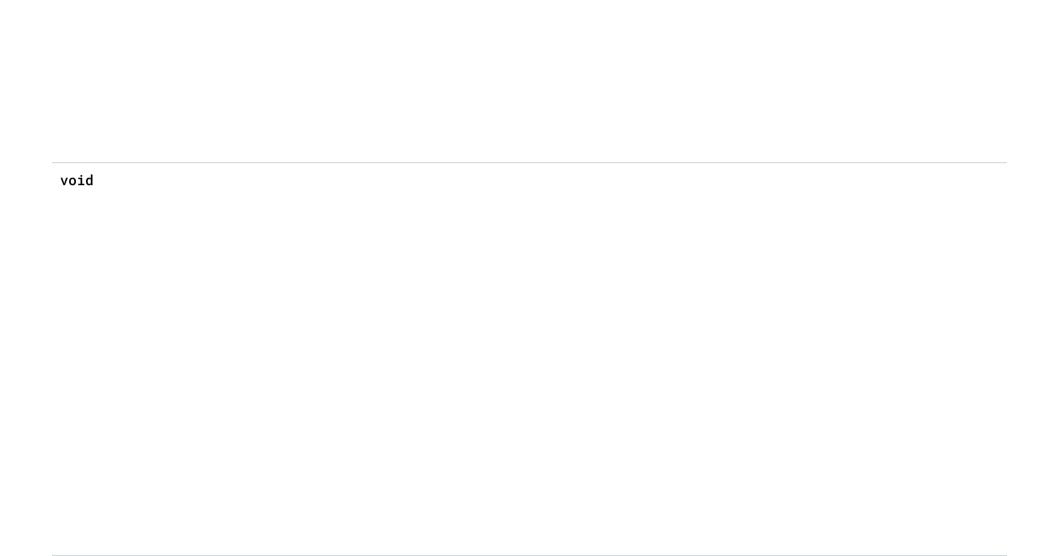


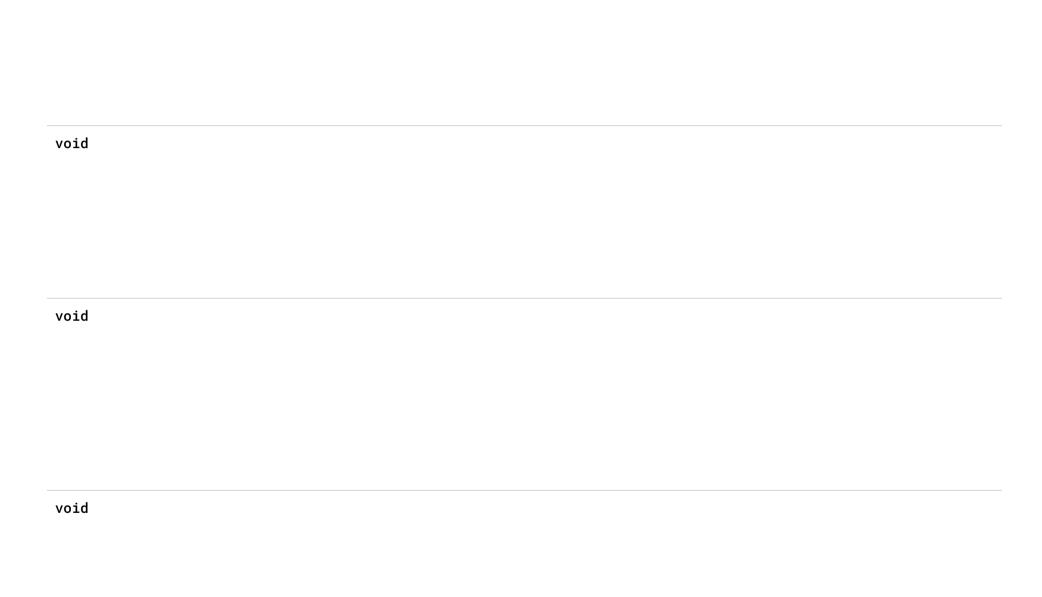


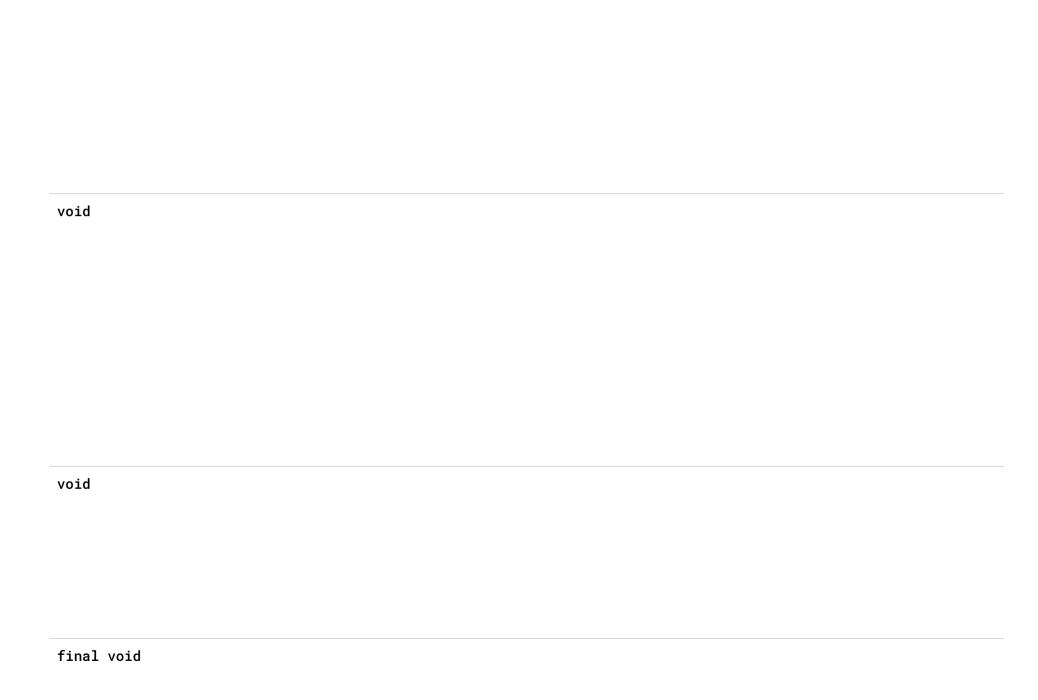


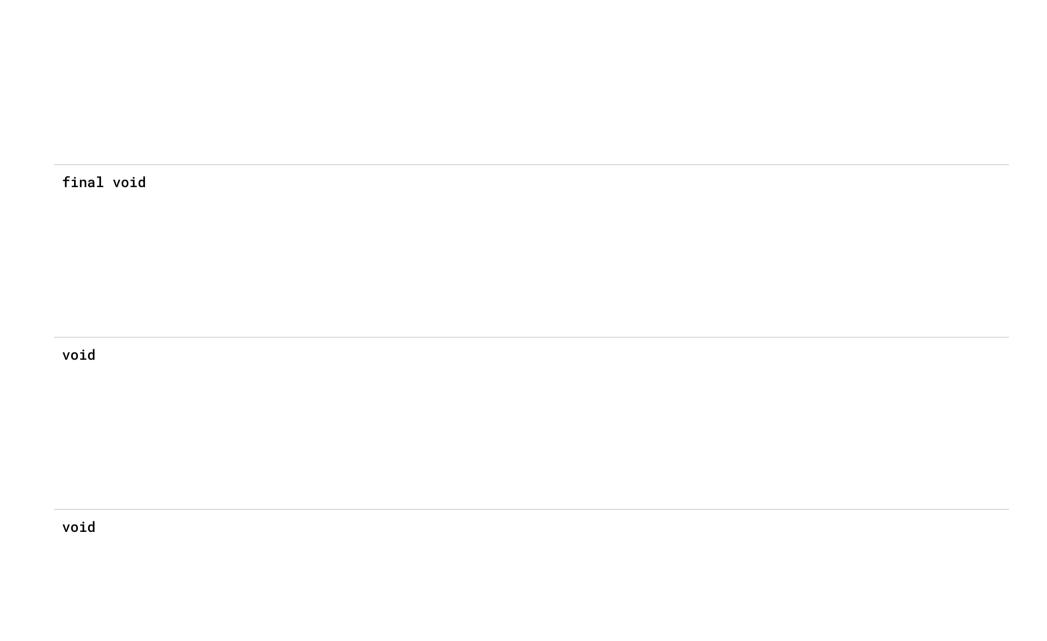


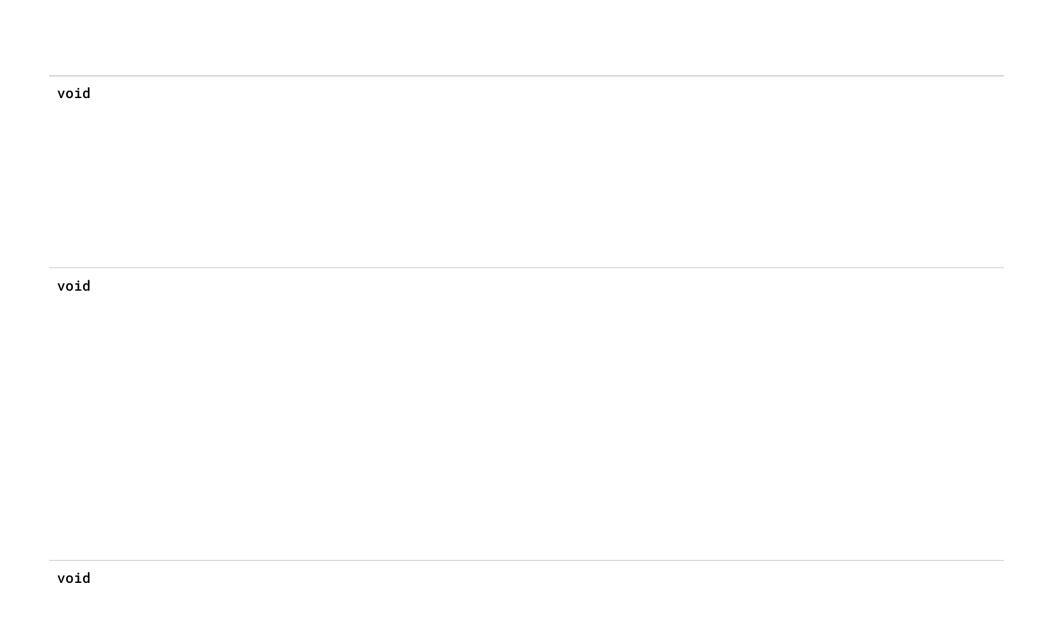


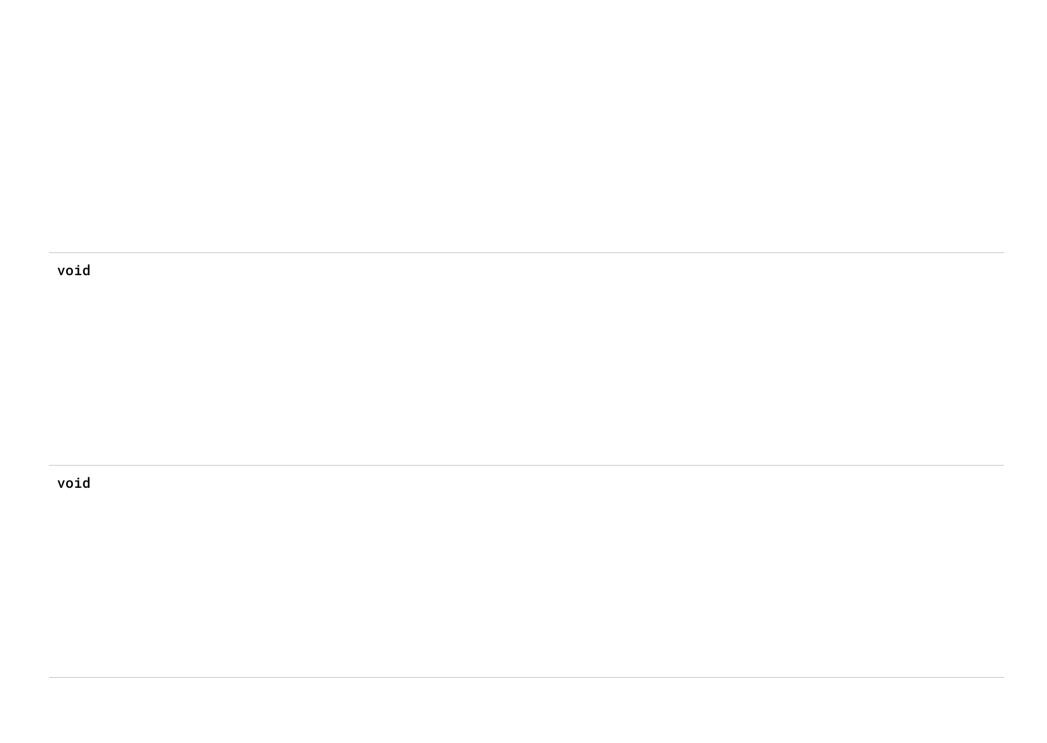






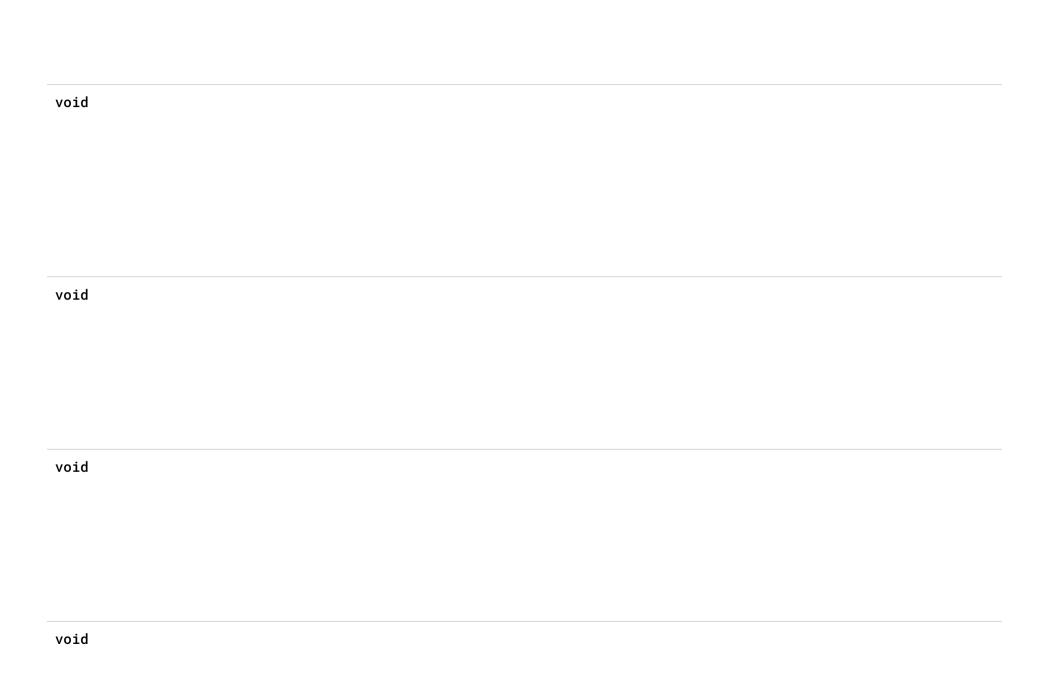


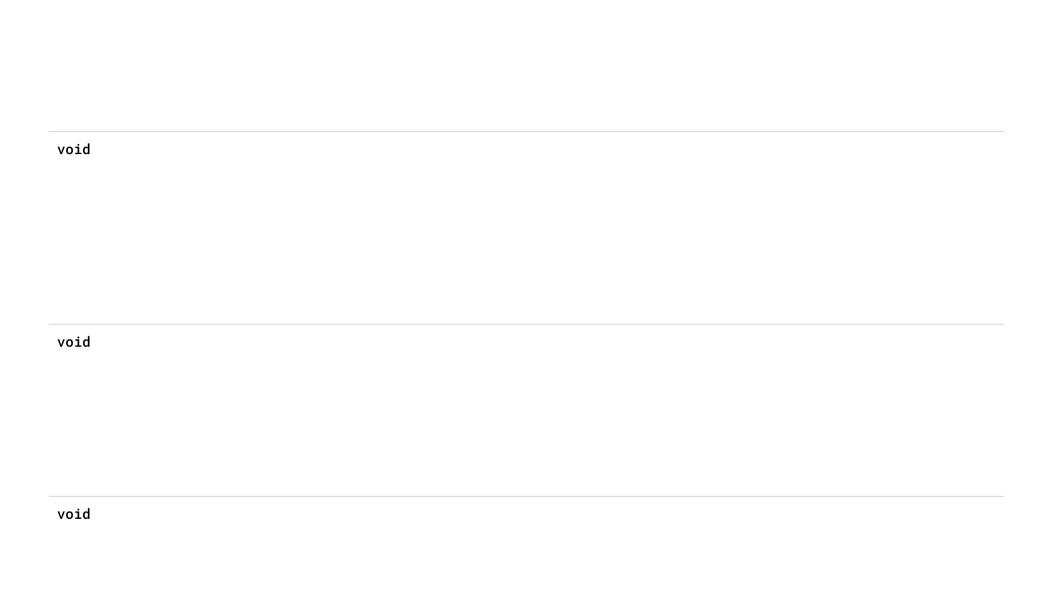


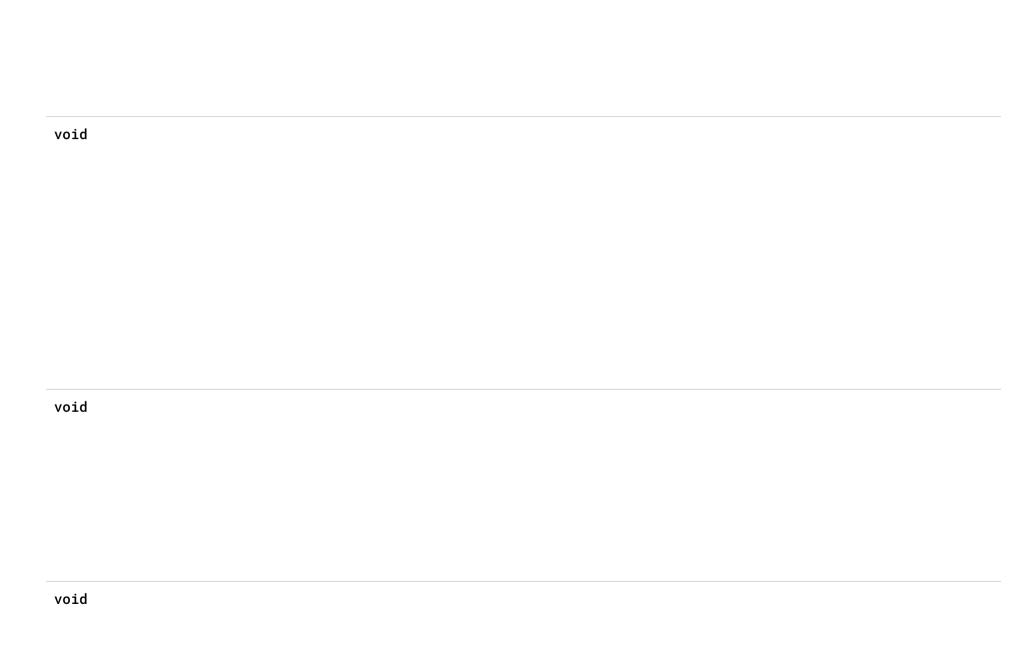


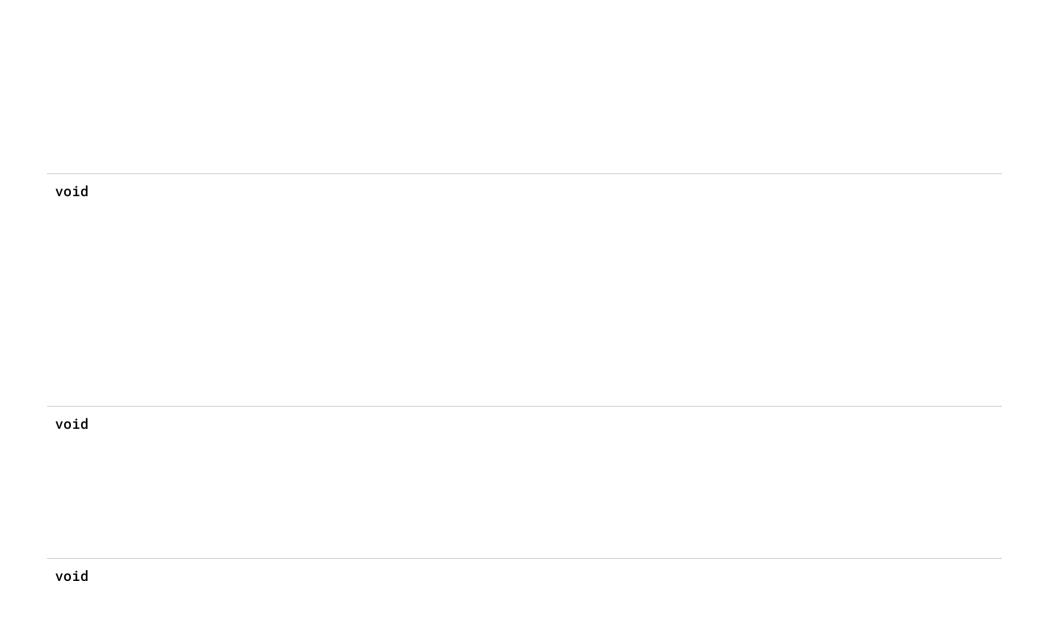


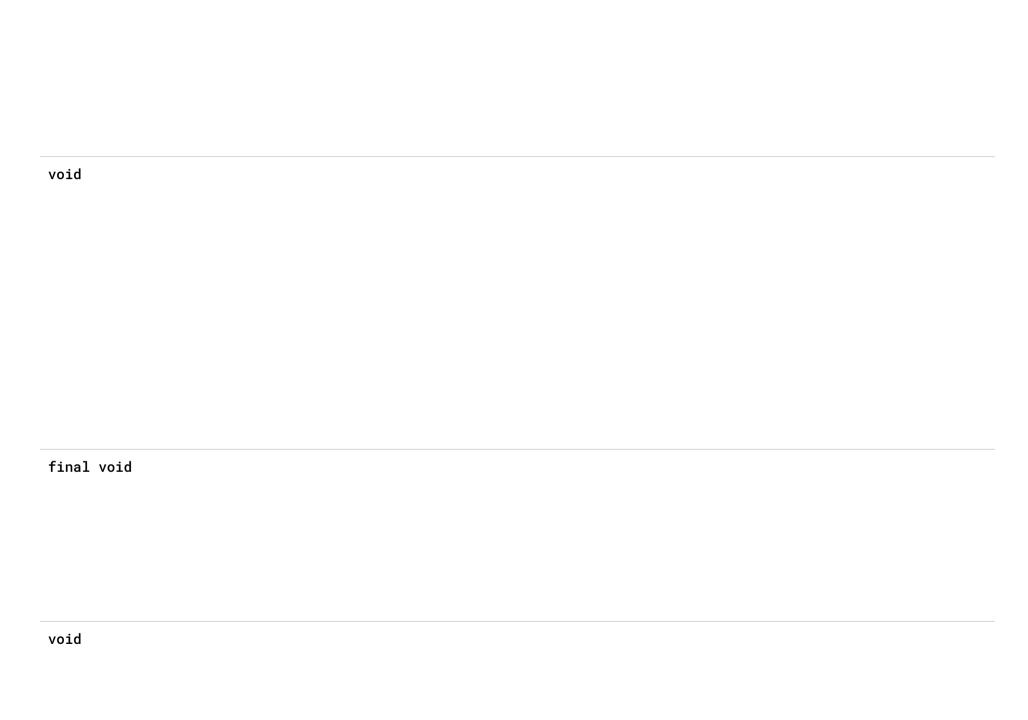
void			
void			
void			
void			

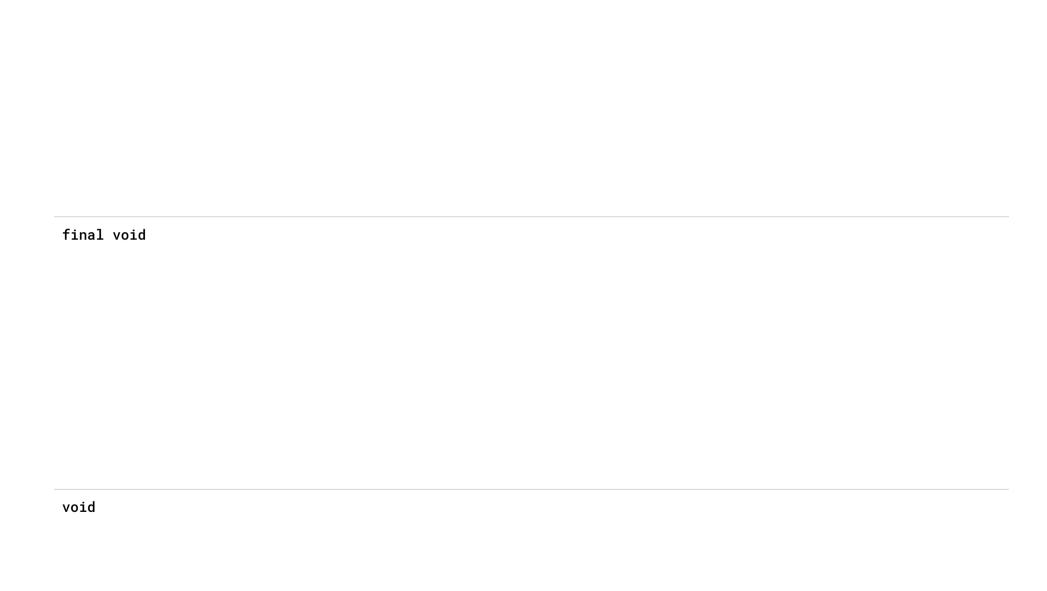


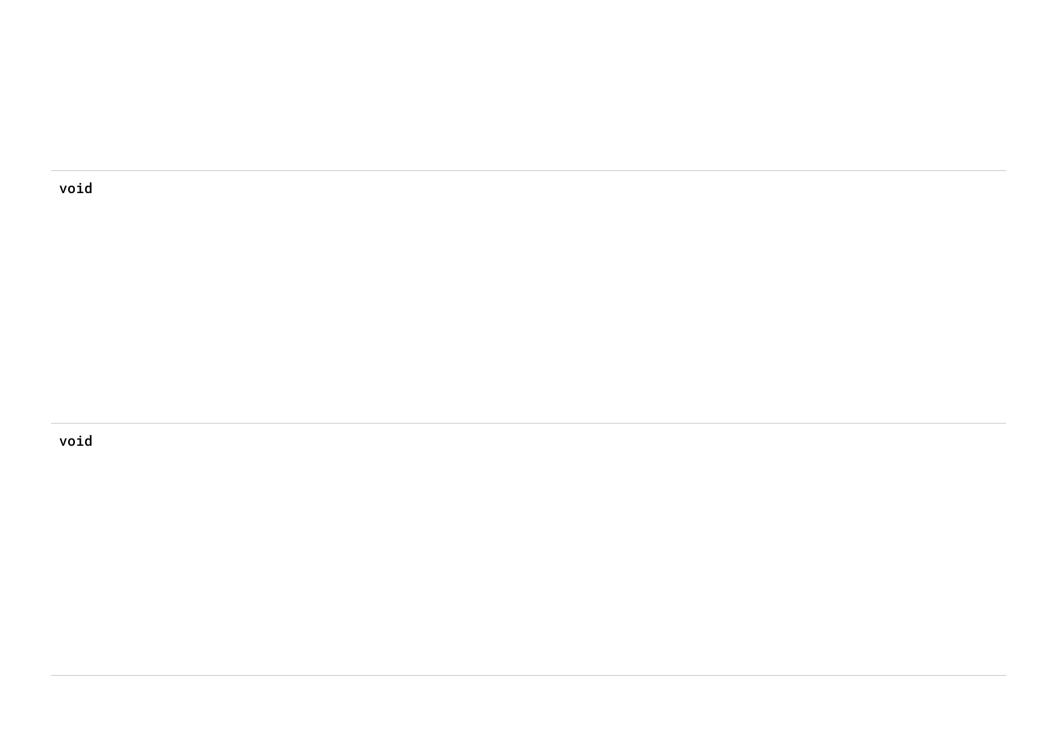


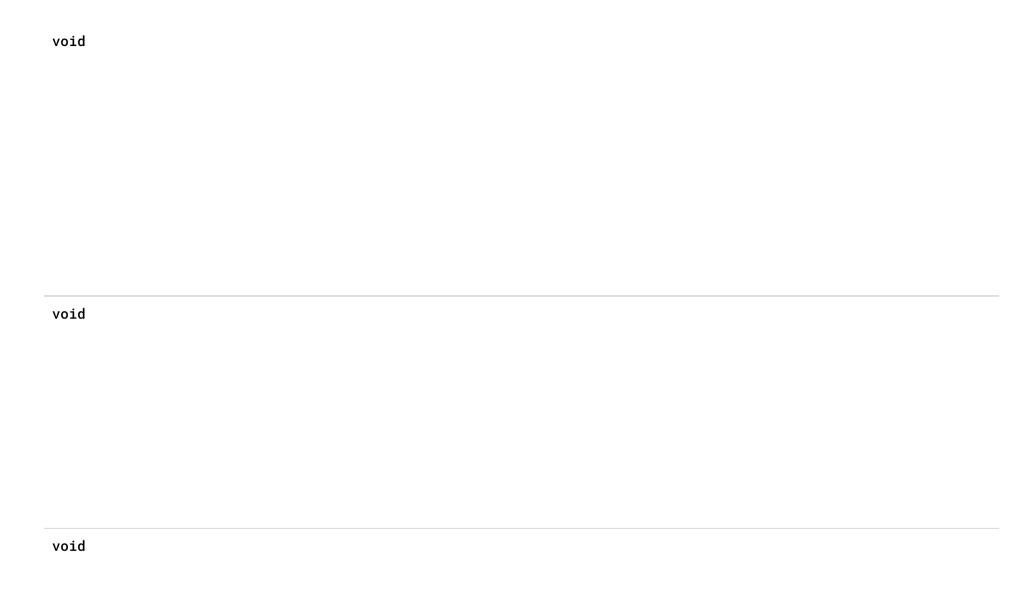


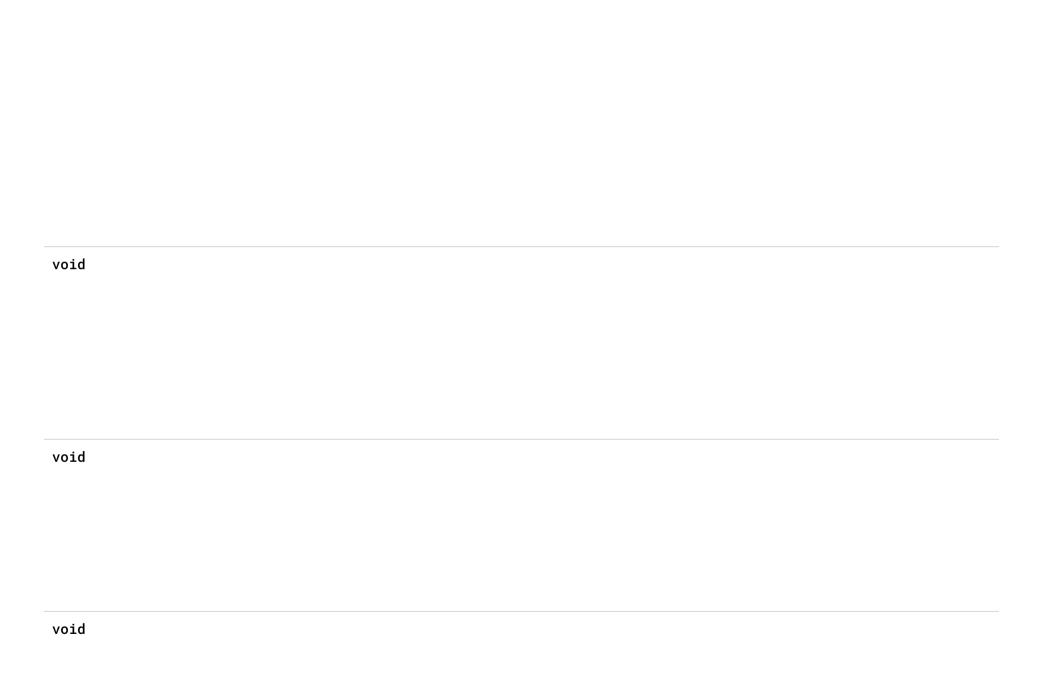


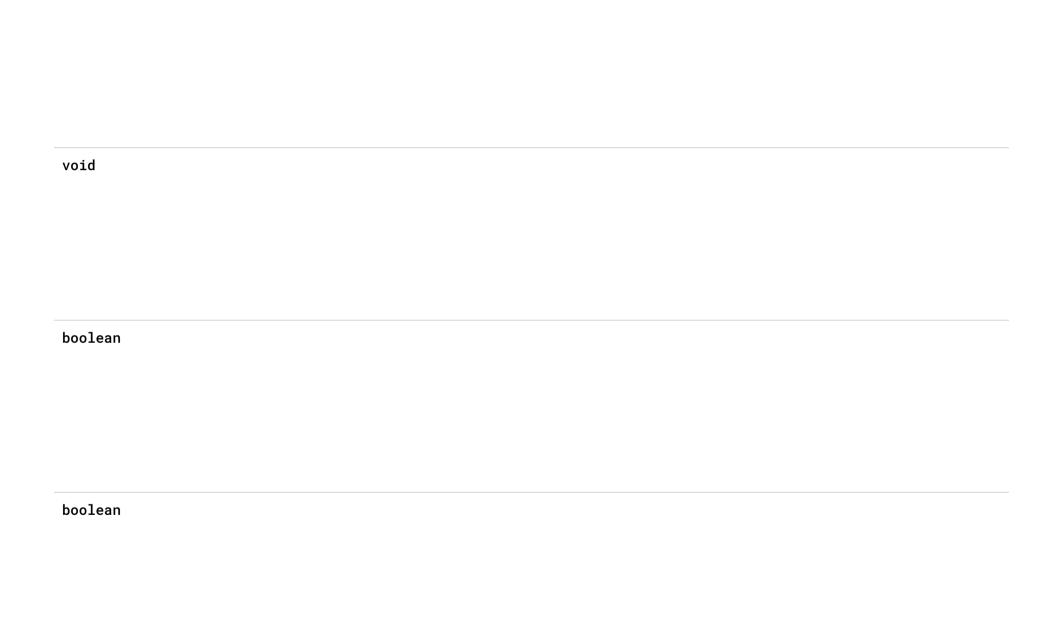


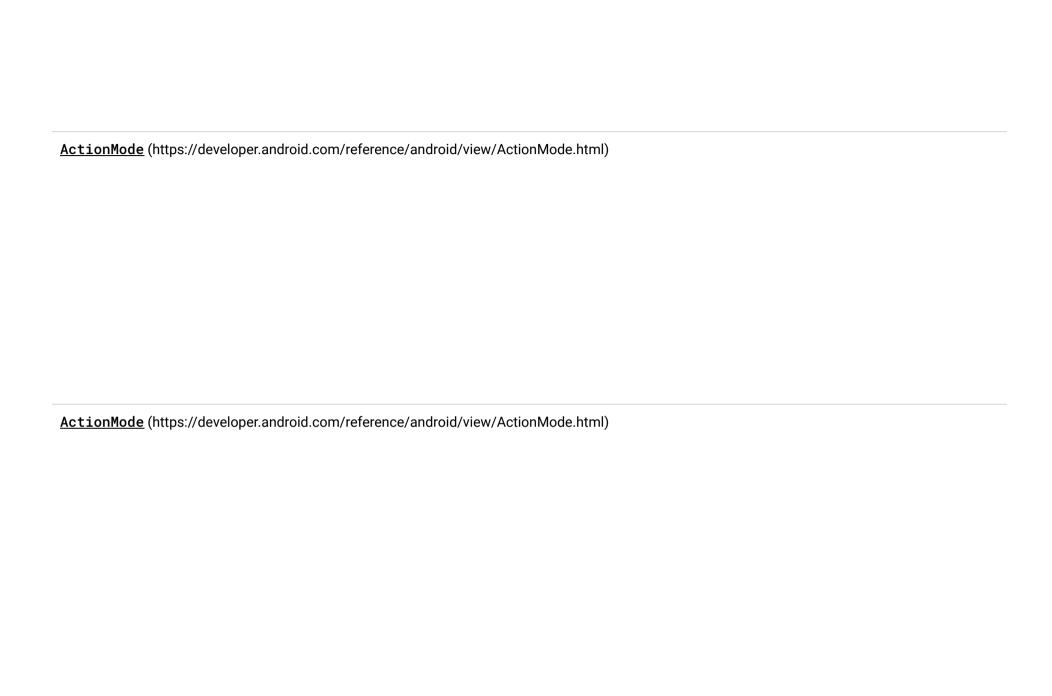






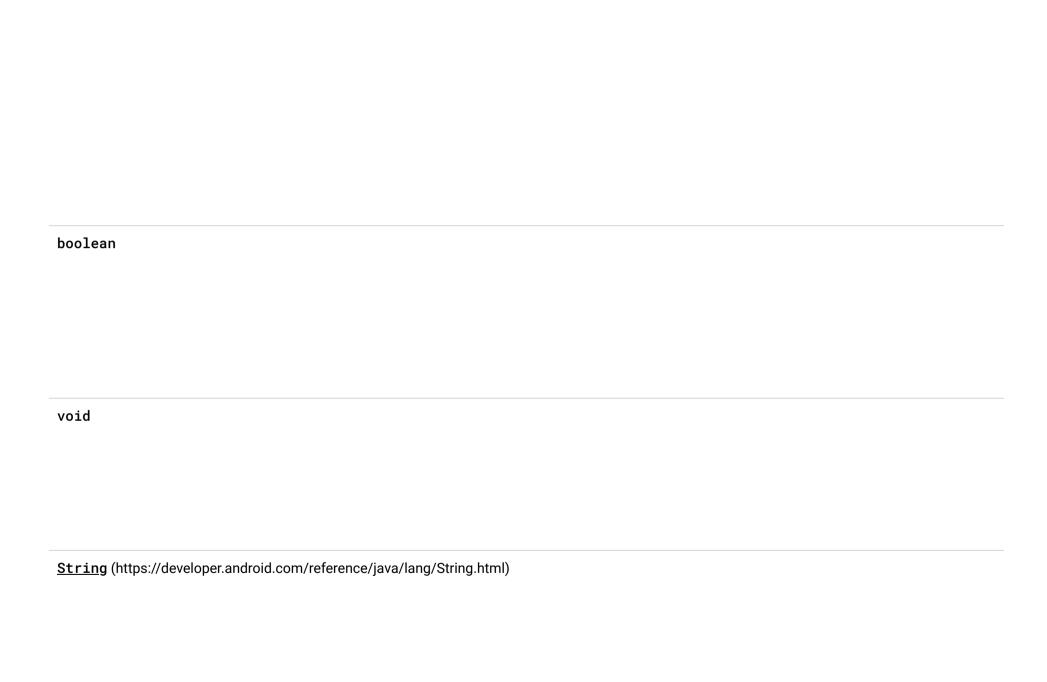






final boolean

final boolean



final void

boolean

boolean

# boolean

From class <u>java.lang.Object</u> (https://developer.android.com/reference/java/la	.9, -2, -3, -3, -3, -3, -3, -3, -3, -3, -3, -3
<u>Object</u> (https://developer.android.com/reference/java/lang/Object.html)	<u>clone</u> (https://developer.android.com/reference/java/lang/Object
	Creates and returns a copy of this object.
boolean	equals (https://developer.android.com/reference/java/lang/Object
	( <u>Object</u> (https://developer.android.com/reference/java/lang/Object
	Indicates whether some other object is "equal to" this one.
void	<u>finalize</u> (https://developer.android.com/reference/java/lang/Obje
	Called by the garbage collector on an object when garbage collectio references to the object.
final Class (https://developer.android.com/reference/java/lang/Class.html)	getClass (https://developer.android.com/reference/java/lang/Obje
	Returns the runtime class of this <b>0bject</b> .

int	hashCode (https://developer.android.com/reference/java/lang/Obje
	Returns a hash code value for the object.
final void	notify (https://developer.android.com/reference/java/lang/Object
	Wakes up a single thread that is waiting on this object's monitor.
final void	notifyAll (https://developer.android.com/reference/java/lang/Ob
	Wakes up all threads that are waiting on this object's monitor.
String (https://developer.android.com/reference/java/lang/String.html)	toString (https://developer.android.com/reference/java/lang/Obje
	Returns a string representation of the object.
final void	<pre>wait (https://developer.android.com/reference/java/lang/Object.htm millis, int nanos)</pre>
	Causes the current thread to wait until another thread invokes the not (https://developer.android.com/reference/java/lang/Object.html#nothers://developer.android.com/reference/java/lang/Object.html#nothers.com/reference/java/lang/Object.html#
final void	wait (https://developer.android.com/reference/java/lang/Object.ht
	Causes the current thread to wait until either another thread invokes (https://developer.android.com/reference/java/lang/Object.html#nc (https://developer.android.com/reference/java/lang/Object.html#nc specified amount of time has elapsed.
final void	wait (https://developer.android.com/reference/java/lang/Object.h
	Causes the current thread to wait until another thread invokes the <u>r</u> (https://developer.android.com/reference/java/lang/Object.html#r

notifyAll() (https://developer.android.com/reference/java/lanç for this object.

From interface android.graphics.drawable.Drawable.Callback (https://developer.android.com/reference/android/graphics/drawable/Drawab

#### abstract void

#### <u>invalidateDrawable</u>

Called when the drawable needs to be redrawn.

#### abstract void

#### scheduleDrawable

(https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable(andawa.lang.Runnable,%20long))

(<u>Drawable</u> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html) who, <u>Runnable</u> (https://developer.android.com/reference/java/lang/Runnable.html) what, <u>long when</u>)

A Drawable can call this to schedule the next frame of its animation.

#### abstract void

#### <u>unscheduleDrawable</u>

(https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#unscheduleDrawable(,,%20java.lang.Runnable))

 $\label{lem:com/reference/android/graphics/drawable/Drawable.html) who, $$Runnable$ (https://developer.android.com/reference/java/lang/Runnable.html) what)$ 

A Drawable can call this to unschedule an action previously scheduled with <a href="mailto:scheduleDrawable">scheduleDrawable</a> (Drawable, Runna (https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable(and 20java.lang.Runnable,%20long))

.

From interface <a href="mailto:android.view.KeyEvent.Callback">android.view.KeyEvent.Callback</a>. (https://developer.android.com/reference/android/view/KeyEvent.Callback.html)

abstract boolean

onKeyDown (https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int,%20andr keyCode, KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) event)

	Called when a key down event has occurred.
abstract boolean	onKeyLongPress (https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyLongPress(ir keyCode, KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) event)
	Called when a long press has occurred.
abstract boolean	<pre>onKeyMultiple (https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyMultiple(int,%2) (int keyCode, int count, KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) e</pre>
	Called when a user's interaction with an analog control, such as flinging a trackball, generates simulated down/up events quick succession.
abstract boolean	onKeyUp (https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyUp(int,%20android.view/KeyEvent.html) event)
	Called when a key up event has occurred.
From interface android.	view.accessibility.AccessibilityEventSource (https://developer.android.com/reference/android/view/accessibili
abstract void	<u>sendAccessibilityEvent</u>
	(https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html#sendAccessibility
	Handles the request for sending an <u>AccessibilityEvent</u> (https://developer.android.com/reference/android/view/a given the event type.
abstract void	<u>sendAccessibilityEventUnchecked</u>
	(https://developer.android.com/reference/android/view/accessibility/AccessibilityEventSource.html#sendAccessibility essibility.AccessibilityEvent))
	(AccessibilityEvent (https://developer.android.com/reference/android/view/accessibility/AccessibilityEvent.htm
	Handles the request for sending an <u>AccessibilityEvent</u> (https://developer.android.com/reference/android/view/a

From interface android.wic	<u>lget.MediaController.MediaPlayerControl</u> (https://developer.android.com/reference/android/widget/MediaControl
abstract boolean	canPause (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#canPa
abstract boolean	$\underline{\textbf{canSeekBackward}} \ (\text{https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.htm}) \\$
abstract boolean	<u>canSeekForward</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html
abstract int	$\underline{\texttt{getAudioSessionId}} \ (\text{https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.log}) \ (\text{https://developer.android/widget/MediaController.MediaPlayerControl.log}) \ (\text{https://developer.android/widget/MediaController.MediaPlayerControl.log}) \ (\text{https://developer.android/widget/MediaController.MediaPlayerControl.log}) \ (\text{https://developer.android/widget/MediaController.MediaPlayerControl.log}) \ (https://developer.android/widget/MediaController.MediaPlayerCont$
	Get the audio session id for the player used by this VideoView.
abstract int	$\underline{\texttt{getBufferPercentage}} \ (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerController.MediaP$
abstract int	$\underline{\texttt{getCurrentPosition}} \ (\text{https://developer.android.com/reference/android/widget/MediaController.MediaPlayerController.}) \\$
abstract int	getDuration (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#ge
abstract boolean	<u>isPlaying</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#isPlay
abstract void	<u>pause</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#pause())()
abstract void	<u>seekTo</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#seekTo(in
abstract void	<u>start</u> (https://developer.android.com/reference/android/widget/MediaController.MediaPlayerControl.html#start())()

# Public constructors

public VideoView (Context (https://developer.android.com/reference/android/content/Context.html) context)

## **Parameters**

context

Context

# VideoView

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

### **Parameters**

context	Context
attrs	AttributeSet

## **VideoView**

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

## **Parameters**

context	Context
attrs	AttributeSet
defStyleAttr	int

# VideoView

added in API level 21 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

# **Parameters**

context	Context
attrs	AttributeSet
defStyleAttr	int
defStyleRes	int

# Public methods

## addSubtitleSource

added in API level 19 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Adds an external subtitle source file (from the provided input stream.) Note that a single external subtitle source may contain multiple or no supported tracks in it. If the source contained at least one track in it, one will receive an <a href="MediaPlayer.MEDIA\_INFO\_METADATA\_UPDATE">MediaPlayer.MEDIA\_INFO\_METADATA\_UPDATE</a> (https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA\_INFO\_METADATA\_UPDATE) info message. Otherwise, if reading the source takes excessive time, one will receive a <a href="MediaPlayer.MEDIA\_INFO\_SUBTITLE\_TIMED\_OUT">MediaPlayer.MEDIA\_INFO\_SUBTITLE\_TIMED\_OUT</a>

(https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA\_INFO\_SUBTITLE\_TIMED\_OUT) message. If the source contained no supported track (including an empty source file or null input stream), one will receive a

### MediaPlayer.MEDIA\_INFO\_UNSUPPORTED\_SUBTITLE

(https://developer.android.com/reference/android/media/MediaPlayer.html#MEDIA\_INFO\_UNSUPPORTED\_SUBTITLE) message. One can find the total number of available tracks using <a href="MediaPlayer.getTrackInfo">MediaPlayer.getTrackInfo</a>()

(https://developer.android.com/reference/android/media/MediaPlayer.html#getTrackInfo()) to see what additional tracks become available after this method call.

### **Parameters**

InputStream: input stream containing the subtitle data. It will be closed by the media framework.

format	MediaFormat: the format of the subtitle track(s). Must contain at least the mime type (MediaFormat.KEY_MIME (https://developer.android.com/reference/android/media/MediaFormat.html#KEY_MIME)) and the language (MediaFormat.KEY_LANGUAGE (https://developer.android.com/reference/android/media/MediaFormat.html#KEY_LANGUAGE)) of the file. If the file itself contains the language information, specify "und" for the language.
canPause  public boolean canPause	added in <u>API level 5</u> (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)
Returns	

# canSeekBackward

added in API level 5 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public boolean canSeekBackward ()

## Returns

boolean

boolean

### canSeekForward

added in API level 5 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public boolean canSeekForward ()

#### **Returns**

boolean

### draw

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void draw (Canvas (https://developer.android.com/reference/android/graphics/Canvas.html) canvas)

Manually render this view (and all of its children) to the given Canvas. The view must have already done a full layout before this function is called. When implementing a view, implement <a href="mailto:onDraw(android.graphics.Canvas">onDraw(android.graphics.Canvas</a>)

(https://developer.android.com/reference/android/view/View.html#onDraw(android.graphics.Canvas)) instead of overriding this method. If you do need to override this method, call the superclass version.

### **Parameters**

canvas

Canvas: The Canvas to which the View is rendered.

getAccessibilityClassName

added in API level 23 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public CharSequence (https://developer.android.com/reference/java/lang/CharSequence.html) getAccessibilityClassName ()

Return the class name of this object to be used for accessibility purposes. Subclasses should only override this if they are implementing something that should be seen as a completely new class of view when used by accessibility, unrelated to the class it is deriving from. This is used to fill in AccessibilityNodeInfo.setClassName

(https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#setClassName(java.lang.CharSequence)).

#### Returns

#### CharSequence

(https://developer.android.com/re ference/java/lang/CharSequence. html)

## getAudioSessionId

added in API level 18 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public int getAudioSessionId ()

Get the audio session id for the player used by this VideoView. This can be used to apply audio effects to the audio track of a video.

#### Returns

int

The audio session, or 0 if there was an error.

getBufferPercentage	added in <u>API level 1</u> (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)
<pre>public int getBufferPercentage ()</pre>	
Returns	
int	
getCurrentPosition	added in <u>API level 1</u> (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)
<pre>public int getCurrentPosition ()</pre>	
Returns	
int	
getDuration	added in <u>API level 1</u> (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)
<pre>public int getDuration ()</pre>	
Returns	
int	

# isPlaying

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public boolean isPlaying ()

#### **Returns**

boolean

# onKeyDown

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public boolean onKeyDown (int keyCode,

KeyEvent (https://developer.android.com/reference/android/view/KeyEvent.html) event)

Default implementation of <a href="KeyEvent.Callback.onKeyDown()">KeyEvent.Callback.onKeyDown()</a>

(https://developer.android.com/reference/android/view/KeyEvent.Callback.html#onKeyDown(int,%20android.view.KeyEvent)): perform press of the view when <a href="mailto:KeyEvent.KEYCODE\_DPAD\_CENTER">KeyEvent.KEYCODE\_DPAD\_CENTER</a> (https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE\_ENTER) or <a href="mailto:KeyEvent.KEYCODE\_ENTER">KeyEvent.KEYCODE\_ENTER</a> (https://developer.android.com/reference/android/view/KeyEvent.html#KEYCODE\_ENTER) is released, if the view is enabled and clickable.

Key presses in software keyboards will generally NOT trigger this listener, although some may elect to do so in some situations. Do not rely on this to catch software key presses.

#### **Parameters**

keyCode

int: a key code that represents the button pressed, from <u>KeyEvent</u>
(https://developer.android.com/reference/android/view/KeyEvent.html)

event

KeyEvent: the KeyEvent object that defines the button action

### **Returns**

boolean

If you handled the event, return true. If you want to allow the event to be handled by the next receiver, return false.

### onTouchEvent

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public boolean onTouchEvent (MotionEvent (https://developer.android.com/reference/android/view/MotionEvent.html) ev)
Implement this method to handle touch screen motion events.

If this method is used to detect click actions, it is recommended that the actions be performed by implementing and calling performClick() (https://developer.android.com/reference/android/view/View.html#performClick()). This will ensure consistent system behavior, including:

- · obeying click sound preferences
- dispatching OnClickListener calls
- handling <u>ACTION\_CLICK</u> (https://developer.android.com/reference/android/view/accessibility/AccessibilityNodeInfo.html#ACTION\_CLICK) when accessibility features are enabled

#### **Parameters**

MotionEvent: The motion event.

### **Returns**

boolean

True if the event was handled, false otherwise.

### onTrackballEvent

added in API level 1 (https://developer.android.com/quide/topics/manifest/uses-sdk-element.html#ApiLevels)

public boolean onTrackballEvent (MotionEvent (https://developer.android.com/reference/android/view/MotionEvent.html) ev)

Implement this method to handle trackball motion events. The *relative* movement of the trackball since the last event can be retrieve with <a href="MotionEvent.getX">MotionEvent.getX()</a> (https://developer.android.com/reference/android/view/MotionEvent.html#getX()) and <a href="MotionEvent.getY">MotionEvent.getY()</a>. (https://developer.android.com/reference/android/view/MotionEvent.html#getY()). These are normalized so that a movement of 1 corresponds to the user pressing one DPAD key (so they will often be fractional values, representing the more fine-grained movement information available from a trackball).

#### **Parameters**

ev

MotionEvent: The motion event.

## Returns

boolean

True if the event was handled, false otherwise.

## pause

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void pause ()

# resolveAdjustedSize

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

### **Parameters**

desiredSize	int		
measureSnec	int		

### Returns

int

## resume

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void resume ()

# seekTo

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void seekTo (int msec)

#### **Parameters**

msec

int

## setAudioAttributes

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void setAudioAttributes (<a href="AudioAttributes"><u>AudioAttributes</u></a> (<a href="https://developer.android.com/reference/android/media/AudioAttributes.html"><u>AudioAttributes</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes.html</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes.html</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes.html</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes.html</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes.html</u></a> (<a href="https://developer.android/media/AudioAttributes.html"><u>AudioAttributes.html</u></a> (<a href="https://developer.android/media/Attributes.html"><u>https://developer.android/media/Attributes.html</u></a> (<a href="https://developer.android/media/Attributes.html"><u>https://developer</u>

Sets the <u>AudioAttributes</u> (https://developer.android.com/reference/android/media/AudioAttributes.html) to be used during the playback of the video.

### **Parameters**

attributes

AudioAttributes: non-null AudioAttributes.

## setAudioFocusRequest

added in API level 26 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void setAudioFocusRequest (int focusGain)

Sets which type of audio focus will be requested during the playback, or configures playback to not request audio focus. Valid values for focus requests are <a href="MulioManager.AUDIOFOCUS\_GAIN">AudioManager.AUDIOFOCUS\_GAIN</a>

(https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS\_GAIN), <u>AudioManager.AUDIOFOCUS\_GAIN\_TRANSIENT</u> (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS\_GAIN\_TRANSIENT),

### <u>AudioManager.AUDIOFOCUS\_GAIN\_TRANSIENT\_MAY\_DUCK</u>

(https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS\_GAIN\_TRANSIENT\_MAY\_DUCK), and

#### <u>AudioManager.AUDIOFOCUS\_GAIN\_TRANSIENT\_EXCLUSIVE</u>

(https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS\_GAIN\_TRANSIENT\_EXCLUSIVE). Or use <a href="mailto:AudioManager.audioFocus\_None">AudioManager.audioFocus\_None</a> (https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS\_NONE) to express that audio focus should not be requested when playback starts. You can for instance use this when playing a silent animation through this class, and you don't want to affect other audio applications playing in the background.

#### **Parameters**

#### focusGain

int: the type of audio focus gain that will be requested, or <u>AudioManager.AUDIOFOCUS\_NONE</u>
(https://developer.android.com/reference/android/media/AudioManager.html#AUDIOFOCUS\_NONE) to disable the use audio focus during playback.

### setMediaController

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void setMediaController (MediaController (https://developer.android.com/reference/android/widget/MediaController.html) controlle

### **Parameters**

controller

MediaController

# setOnCompletionListener

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void setOnCompletionListener (MediaPlayer.OnCompletionListener (https://developer.android.com/reference/android/media/Media

Register a callback to be invoked when the end of a media file has been reached during playback.

#### **Parameters**

1 MediaPlayer.OnCompletionListener: The callback that will be run

### setOnErrorListener

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

 $public \ void \ set 0 n Error Listener \ (\underline{\texttt{MediaPlayer.OnError Listener}} \ (\text{https://developer.android.com/reference/android/media/MediaPlayer.OnError Listener}) \ (\underline{\texttt{MediaPlayer.OnError Listener}} \ (\underline{\texttt{MediaPlayer.OnError Listener}} \ (\text{https://developer.android.com/reference/android/media/MediaPlayer.OnError Listener}) \ (\underline{\texttt{MediaPlayer.OnError Listener}} \$ 

Register a callback to be invoked when an error occurs during playback or setup. If no listener is specified, or if the listener returned false, VideoView will inform the user of any errors.

## **Parameters**

1

MediaPlayer.OnErrorListener: The callback that will be run

## setOnInfoListener

added in API level 17 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

 $public \ void \ set On Info Listener \ (\underline{\texttt{MediaPlayer.OnInfo Listener}} \ (\text{https://developer.android.com/reference/android/media/MediaPlayer.OnInfo Listener}) \ (\underline{\texttt{MediaPlayer.OnInfo Listener}} \ (\underline{\texttt{MediaPlayer.OnInfo List$ 

Register a callback to be invoked when an informational event occurs during playback or setup.

### **Parameters**

1 MediaPlayer.OnInfoListener: The callback that will be run

# setOnPreparedListener

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

 $public \ void \ set 0 n Prepared Listener \ (\underline{MediaPlayer.0n Prepared Listener} \ (https://developer.android.com/reference/android/media/MediaPlayer.0n Prepared Listener) \ (\underline{MediaPlayer.0n Prepared Listener)} \ (\underline{MediaPlayer.0n Prepared Lis$ 

Register a callback to be invoked when the media file is loaded and ready to go.

### **Parameters**

1 MediaPlayer.OnPreparedListener: The callback that will be run

setVideoPath

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void setVideoPath (String (https://developer.android.com/reference/java/lang/String.html) path)
Sets video path.

### **Parameters**

path

**String**: the path of the video.

## setVideoURI

added in API level 21 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void setVideoURI (<u>Uri</u> (https://developer.android.com/reference/android/net/Uri.html) uri,

<u>Map</u> (https://developer.android.com/reference/java/util/Map.html)<a href="mailto:string"><u>String</u> (https://developer.android.com/reference/java/lang/String.html)</a>

Sets video URI using specific headers.

## **Parameters**

uri	Uri: the URI of the video.
headers	Map: the headers for the URI request. Note that the cross domain redirection is allowed by default, but that can be changed with key/value pairs through the headers parameter with "android-allow-cross-domain-redirect" as the key and "0" or "1" as the value to disallow or allow cross domain redirection.

## setVideoURI

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void setVideoURI (Uri (https://developer.android.com/reference/android/net/Uri.html) uri)

Sets video URI.

### **Parameters**

uri

Uri: the URI of the video.

## start

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void start ()

# stopPlayback

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void stopPlayback ()

## suspend

added in API level 8 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

public void suspend ()

# Protected methods

## onAttachedToWindow

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

protected void onAttachedToWindow ()

This is called when the view is attached to a window. At this point it has a Surface and will start drawing. Note that this function is guaranteed to be called before <a href="mailto:onDraw(android.graphics.Canvas">onDraw(android.graphics.Canvas</a>)

(https://developer.android.com/reference/android/view/View.html#onDraw(android.graphics.Canvas)), however it may be called any time before the first onDraw -- including before or after <a href="mailto:onMeasure(int, int)">onMeasure(int, int)</a>

(https://developer.android.com/reference/android/view/View.html#onMeasure(int,%20int)).

## onDetachedFromWindow

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

protected void onDetachedFromWindow ()

This is called when the view is detached from a window. At this point it no longer has a surface for drawing.

# onLayout

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

int right,
int bottom)

Called from layout when this view should assign a size and position to each of its children. Derived classes with children should override this method and call layout on each of their children.

### **Parameters**

changed	boolean: This is a new size or position for this view
left	int: Left position, relative to parent
top	int: Top position, relative to parent
right	int: Right position, relative to parent
bottom	int: Bottom position, relative to parent

## onMeasure

added in API level 1 (https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels)

Measure the view and its content to determine the measured width and the measured height. This method is invoked by <a href="measure(int,">measure(int,</a> (https://developer.android.com/reference/android/view/View.html#measure(int,%20int)) and should be overridden by subclasses to provide

accurate and efficient measurement of their contents.

**CONTRACT:** When overriding this method, you *must* call <u>setMeasuredDimension(int, int)</u>

(https://developer.android.com/reference/android/view/View.html#setMeasuredDimension(int,%20int)) to store the measured width and height of this view. Failure to do so will trigger an IllegalStateException, thrown by <a href="mailto:measure(int, int)">measure(int, int)</a>

(https://developer.android.com/reference/android/view/View.html#measure(int,%20int)). Calling the superclass' <a href="mailto:onMeasure(int, int">onMeasure(int</a>, <a href="mailto:int">int</a>) (https://developer.android.com/reference/android/view/View.html#onMeasure(int,%20int)) is a valid use.

The base class implementation of measure defaults to the background size, unless a larger size is allowed by the MeasureSpec. Subclasses should override <a href="mailto:onMeasure(int, int">onMeasure(int, int)</a> (https://developer.android.com/reference/android/view/View.html#onMeasure(int,%20int)) to provide better measurements of their content.

If this method is overridden, it is the subclass's responsibility to make sure the measured height and width are at least the view's minimum height and width (getSuggestedMinimumHeight()

 $(https://developer.android.com/reference/android/view/View.html\#getSuggestedMinimumHeight()) \ and \ \underline{getSuggestedMinimumWidth()} \\ (https://developer.android.com/reference/android/view/View.html\#getSuggestedMinimumWidth())).$ 

#### **Parameters**

widthMeasureSpec	int: horizontal space requirements as imposed by the parent. The requirements are encoded with <u>View.MeasureSpec</u> (https://developer.android.com/reference/android/view/View.MeasureSpec.html).
heightMeasureSpec	int: vertical space requirements as imposed by the parent. The requirements are encoded with <u>View.MeasureSpec</u> (https://developer.android.com/reference/android/view/View.MeasureSpec.html).

Content and code samples on this page are subject to the licenses described in the <u>Content License</u> (/license). Java is a registered trademark of Oracle and/or its affiliates.

Last updated June 6, 2018.



<u>Twitter</u> Follow @AndroidDev on Twitter



Google+
Follow Android Developers on
Google+



<u>YouTube</u> Check out Android Developers on YouTube