

public static final
enum

Summary: [Enums](#) | [Methods](#) | [Inherited Methods](#) | [\[Expand All\]](#)

Added in [API level 1](#)

ImageView.ScaleType

extends [Enum](#)<E extends [Enum](#)<E>>

[java.lang.Object](#)

↳ [java.lang.Enum](#)<E extends [java.lang.Enum](#)<E>>

↳ android.widget.ImageView.ScaleType

Class Overview

Options for scaling the bounds of an image to the bounds of this view.

Summary

Enum Values	
ImageView.ScaleType CENTER	Center the image in the view, but perform no scaling.
ImageView.ScaleType CENTER_CROP	Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or larger than the corresponding dimension of the view (minus padding).
ImageView.ScaleType CENTER_INSIDE	Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or less than the corresponding dimension of the view (minus padding).
ImageView.ScaleType FIT_CENTER	Scale the image using CENTER.
ImageView.ScaleType FIT_END	Scale the image using END.
ImageView.ScaleType FIT_START	Scale the image using START.
ImageView.ScaleType FIT_XY	Scale the image using FILL.

ImageView.ScaleType MATRIX Scale using the image matrix when drawing.

Public Methods

static ImageView.ScaleType valueOf(String name)
final static ScaleType[] values()

Inherited Methods [Expand]

- ▶ From class java.lang.Enum
- ▶ From class java.lang.Object
- ▶ From interface java.lang.Comparable

Enum Values

public static final ImageView.ScaleType **CENTER** Added in API level 1

Center the image in the view, but perform no scaling. From XML, use this syntax: android:scaleType="center".

public static final ImageView.ScaleType **CENTER_CROP** Added in API level 1

Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or larger than the corresponding dimension of the view (minus padding). The image is then centered in the view. From XML, use this syntax: android:scaleType="centerCrop".

public static final ImageView.ScaleType **CENTER_INSIDE** Added in API level 1

Scale the image uniformly (maintain the image's aspect ratio) so that both dimensions (width and height) of the image will be equal to or less than the corresponding dimension of the view (minus padding). The image is then centered in the view. From XML, use this syntax: android:scaleType="centerInside".

public static final ImageView.ScaleType **FIT_CENTER** Added in API level 1

Scale the image using CENTER (</reference/android/graphics/Matrix.ScaleToFit.html#CENTER>). From XML, use this syntax: android:scaleType="fitCenter".

public static final ImageView.ScaleType **FIT_END** Added in API level 1

Scale the image using END (</reference/android/graphics/Matrix.ScaleToFit.html#END>). From XML, use this syntax:
`android:scaleType="fitEnd".`

public static final ImageView.ScaleType **FIT_START** Added in API level 1

Scale the image using START (</reference/android/graphics/Matrix.ScaleToFit.html#START>). From XML, use this syntax:
`android:scaleType="fitStart".`

public static final ImageView.ScaleType **FIT_XY** Added in API level 1

Scale the image using FILL (</reference/android/graphics/Matrix.ScaleToFit.html#FILL>). From XML, use this syntax:
`android:scaleType="fitXY".`

public static final ImageView.ScaleType **MATRIX** Added in API level 1

Scale using the image matrix when drawing. The image matrix can be set using `setImageMatrix(Matrix)` ([/reference/android/widget/ImageView.html#setImageMatrix\(android.graphics.Matrix\)](/reference/android/widget/ImageView.html#setImageMatrix(android.graphics.Matrix))). From XML, use this syntax: `android:scaleType="matrix".`

Public Methods

public static ImageView.ScaleType **valueOf** (String name) Added in API level 1

public static final ScaleType[] **values** () Added in API level 1