Flutter resumen documentación

# Assets, Images and Icon widgets

## asset Bundle

A collection of resources used by the application.

Asset bundles contain resources, such as images and strings, that can be used by an application. Access to these resources is asynchronous so that they can be transparently loaded over a network (e.g., from a [NetworkAssetBundle](https://api.flutter.dev/flutter/services/NetworkAssetBundle-class.html)) or from the local file system without blocking the application's user interface.

Applications have a [rootBundle](https://api.flutter.dev/flutter/services/rootBundle.html), which contains the resources that were packaged with the application when it was built. To add resources to the [rootBundle](https://api.flutter.dev/flutter/services/rootBundle.html) for your application, add them to the assets subsection of the flutter section of your application's pubspec.yaml manifest.

For example:

name: my\_awesome\_application

flutter:

assets:

- images/hamilton.jpeg

- images/lafayette.jpeg

Rather than accessing the [rootBundle](https://api.flutter.dev/flutter/services/rootBundle.html) global static directly, consider obtaining the [AssetBundle](https://api.flutter.dev/flutter/services/AssetBundle-class.html) for the current [BuildContext](https://api.flutter.dev/flutter/widgets/BuildContext-class.html) using [DefaultAssetBundle.of](https://api.flutter.dev/flutter/widgets/DefaultAssetBundle/of.html). This layer of indirection lets ancestor widgets substitute a different [AssetBundle](https://api.flutter.dev/flutter/services/AssetBundle-class.html) (e.g., for testing or localization) at runtime rather than directly replying upon the [rootBundle](https://api.flutter.dev/flutter/services/rootBundle.html) created at build time. For convenience, the [WidgetsApp](https://api.flutter.dev/flutter/widgets/WidgetsApp-class.html) or [MaterialApp](https://api.flutter.dev/flutter/material/MaterialApp-class.html) widget at the top of the widget hierarchy configures the [DefaultAssetBundle](https://api.flutter.dev/flutter/widgets/DefaultAssetBundle-class.html) to be the [rootBundle](https://api.flutter.dev/flutter/services/rootBundle.html).

### Constructors

[AssetBundle](https://api.flutter.dev/flutter/services/AssetBundle/AssetBundle.html)()

### Properties

[hashCode](https://api.flutter.dev/flutter/dart-core/Object/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

### Methods

[clear](https://api.flutter.dev/flutter/services/AssetBundle/clear.html)() → void

If this is a caching asset bundle, clear all cached data.

[evict](https://api.flutter.dev/flutter/services/AssetBundle/evict.html)([String](https://api.flutter.dev/flutter/dart-core/String-class.html) key) → void

If this is a caching asset bundle, and the given key describes a cached asset, then evict the asset from the cache so that the next time it is loaded, the cache will be reread from the asset bundle.

[load](https://api.flutter.dev/flutter/services/AssetBundle/load.html)([String](https://api.flutter.dev/flutter/dart-core/String-class.html) key) → [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<[ByteData](https://api.flutter.dev/flutter/dart-typed_data/ByteData-class.html)>

Retrieve a binary resource from the asset bundle as a data stream.

[loadBuffer](https://api.flutter.dev/flutter/services/AssetBundle/loadBuffer.html)([String](https://api.flutter.dev/flutter/dart-core/String-class.html) key) → [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<[ImmutableBuffer](https://api.flutter.dev/flutter/dart-ui/ImmutableBuffer-class.html)>

Retrieve a binary resource from the asset bundle as an immutable buffer.

[loadString](https://api.flutter.dev/flutter/services/AssetBundle/loadString.html)([String](https://api.flutter.dev/flutter/dart-core/String-class.html) key, {[bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) cache = true}) → [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<[String](https://api.flutter.dev/flutter/dart-core/String-class.html)>

Retrieve a string from the asset bundle.

[loadStructuredBinaryData](https://api.flutter.dev/flutter/services/AssetBundle/loadStructuredBinaryData.html)<T>([String](https://api.flutter.dev/flutter/dart-core/String-class.html) key, [FutureOr](https://api.flutter.dev/flutter/dart-async/FutureOr-class.html)<T> parser([ByteData](https://api.flutter.dev/flutter/dart-typed_data/ByteData-class.html) data)) → [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<T>

Retrieve [ByteData](https://api.flutter.dev/flutter/dart-typed_data/ByteData-class.html) from the asset bundle, parse it with the given function, and return that function's result.

[loadStructuredData](https://api.flutter.dev/flutter/services/AssetBundle/loadStructuredData.html)<T>([String](https://api.flutter.dev/flutter/dart-core/String-class.html) key, [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<T> parser([String](https://api.flutter.dev/flutter/dart-core/String-class.html) value)) → [Future](https://api.flutter.dev/flutter/dart-async/Future-class.html)<T>

Retrieve a string from the asset bundle, parse it with the given function, and return that function's result.

[noSuchMethod](https://api.flutter.dev/flutter/dart-core/Object/noSuchMethod.html)([Invocation](https://api.flutter.dev/flutter/dart-core/Invocation-class.html) invocation) → dynamic

Invoked when a nonexistent method or property is accessed.

inherited

[toString](https://api.flutter.dev/flutter/services/AssetBundle/toString.html)() → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)

A string representation of this object

## icon

A graphical icon widget drawn with a glyph from a font described in an [IconData](https://api.flutter.dev/flutter/widgets/IconData-class.html) such as material's predefined [IconData](https://api.flutter.dev/flutter/widgets/IconData-class.html)s in [Icons](https://api.flutter.dev/flutter/material/Icons-class.html).

Icons are not interactive. For an interactive icon, consider material's [IconButton](https://api.flutter.dev/flutter/material/IconButton-class.html).

There must be an ambient [Directionality](https://api.flutter.dev/flutter/widgets/Directionality-class.html) widget when using [Icon](https://api.flutter.dev/flutter/widgets/Icon-class.html). Typically this is introduced automatically by the [WidgetsApp](https://api.flutter.dev/flutter/widgets/WidgetsApp-class.html) or [MaterialApp](https://api.flutter.dev/flutter/material/MaterialApp-class.html).

This widget assumes that the rendered icon is squared. Non-squared icons may render incorrectly.

### Constructors

[Icon](https://api.flutter.dev/flutter/widgets/Icon/Icon.html)([IconData](https://api.flutter.dev/flutter/widgets/IconData-class.html)? icon, {[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? size, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? fill, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? weight, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? grade, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? opticalSize, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? color, [List](https://api.flutter.dev/flutter/dart-core/List-class.html)<[Shadow](https://api.flutter.dev/flutter/dart-ui/Shadow-class.html)>? shadows, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticLabel, [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)? textDirection})

Creates an icon.

const

### Properties

[color](https://api.flutter.dev/flutter/widgets/Icon/color.html) → [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)?

The color to use when drawing the icon.

final

[fill](https://api.flutter.dev/flutter/widgets/Icon/fill.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The fill for drawing the icon.

final

[grade](https://api.flutter.dev/flutter/widgets/Icon/grade.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The grade (granular stroke weight) for drawing the icon.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[icon](https://api.flutter.dev/flutter/widgets/Icon/icon.html) → [IconData](https://api.flutter.dev/flutter/widgets/IconData-class.html)?

The icon to display. The available icons are described in [Icons](https://api.flutter.dev/flutter/material/Icons-class.html).

final

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[opticalSize](https://api.flutter.dev/flutter/widgets/Icon/opticalSize.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The optical size for drawing the icon.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[semanticLabel](https://api.flutter.dev/flutter/widgets/Icon/semanticLabel.html) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)?

Semantic label for the icon.

final

[shadows](https://api.flutter.dev/flutter/widgets/Icon/shadows.html) → [List](https://api.flutter.dev/flutter/dart-core/List-class.html)<[Shadow](https://api.flutter.dev/flutter/dart-ui/Shadow-class.html)>?

A list of Shadows that will be painted underneath the icon.

final

[size](https://api.flutter.dev/flutter/widgets/Icon/size.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The size of the icon in logical pixels.

final

[textDirection](https://api.flutter.dev/flutter/widgets/Icon/textDirection.html) → [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html)?

The text direction to use for rendering the icon.

final

[weight](https://api.flutter.dev/flutter/widgets/Icon/weight.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

The stroke weight for drawing the icon.

## image

A widget that displays an image.

Several constructors are provided for the various ways that an image can be specified:

* [Image.new](https://api.flutter.dev/flutter/widgets/Image/Image.html), for obtaining an image from an [ImageProvider](https://api.flutter.dev/flutter/painting/ImageProvider-class.html).
* [Image.asset](https://api.flutter.dev/flutter/widgets/Image/Image.asset.html), for obtaining an image from an [AssetBundle](https://api.flutter.dev/flutter/services/AssetBundle-class.html) using a key.
* [Image.network](https://api.flutter.dev/flutter/widgets/Image/Image.network.html), for obtaining an image from a URL.
* [Image.file](https://api.flutter.dev/flutter/widgets/Image/Image.file.html), for obtaining an image from a [File](https://api.flutter.dev/flutter/dart-io/File-class.html).
* [Image.memory](https://api.flutter.dev/flutter/widgets/Image/Image.memory.html), for obtaining an image from a [Uint8List](https://api.flutter.dev/flutter/dart-typed_data/Uint8List-class.html).

The following image formats are supported: JPEG, PNG, GIF, Animated GIF, WebP, Animated WebP, BMP, and WBMP. Additional formats may be supported by the underlying platform. Flutter will attempt to call platform API to decode unrecognized formats, and if the platform API supports decoding the image Flutter will be able to render it.

To automatically perform pixel-density-aware asset resolution, specify the image using an [AssetImage](https://api.flutter.dev/flutter/painting/AssetImage-class.html) and make sure that a [MaterialApp](https://api.flutter.dev/flutter/material/MaterialApp-class.html), [WidgetsApp](https://api.flutter.dev/flutter/widgets/WidgetsApp-class.html), or [MediaQuery](https://api.flutter.dev/flutter/widgets/MediaQuery-class.html) widget exists above the [Image](https://api.flutter.dev/flutter/widgets/Image-class.html) widget in the widget tree.

The image is painted using [paintImage](https://api.flutter.dev/flutter/painting/paintImage.html), which describes the meanings of the various fields on this class in more detail.

### Memory usage

The image is stored in memory in uncompressed form (so that it can be rendered). Large images will use a lot of memory: a 4K image (3840×2160) will use over 30MB of RAM (assuming 32 bits per pixel).

This problem is exacerbated by the images being cached in the [ImageCache](https://api.flutter.dev/flutter/painting/ImageCache-class.html), so large images can use memory for even longer than they are displayed.

The [Image.asset](https://api.flutter.dev/flutter/widgets/Image/Image.asset.html), [Image.network](https://api.flutter.dev/flutter/widgets/Image/Image.network.html), [Image.file](https://api.flutter.dev/flutter/widgets/Image/Image.file.html), and [Image.memory](https://api.flutter.dev/flutter/widgets/Image/Image.memory.html) constructors allow a custom decode size to be specified through cacheWidth and cacheHeight parameters. The engine will then decode and store the image at the specified size, instead of the image's natural size.

This can significantly reduce the memory usage. For example, a 4K image that will be rendered at only 384×216 pixels (one-tenth the horizontal and vertical dimensions) would only use 330KB if those dimensions are specified using the cacheWidth and cacheHeight parameters, a 100-fold reduction in memory usage.

### Web considerations

In the case where a network image is used on the Web platform, the cacheWidth and cacheHeight parameters are only supported when the application is running with the CanvasKit renderer. When the application is using the HTML renderer, the web engine delegates image decoding of network images to the Web, which does not support custom decode sizes.

### Constructors

[Image](https://api.flutter.dev/flutter/widgets/Image/Image.html)({[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, required [ImageProvider](https://api.flutter.dev/flutter/painting/ImageProvider-class.html)<[Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)> image, [ImageFrameBuilder](https://api.flutter.dev/flutter/widgets/ImageFrameBuilder.html)? frameBuilder, [ImageLoadingBuilder](https://api.flutter.dev/flutter/widgets/ImageLoadingBuilder.html)? loadingBuilder, [ImageErrorWidgetBuilder](https://api.flutter.dev/flutter/widgets/ImageErrorWidgetBuilder.html)? errorBuilder, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticLabel, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeFromSemantics = false, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? width, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? height, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? color, [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>? opacity, [BlendMode](https://api.flutter.dev/flutter/dart-ui/BlendMode.html)? colorBlendMode, [BoxFit](https://api.flutter.dev/flutter/painting/BoxFit.html)? fit, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment = Alignment.center, [ImageRepeat](https://api.flutter.dev/flutter/painting/ImageRepeat.html) repeat = ImageRepeat.noRepeat, [Rect](https://api.flutter.dev/flutter/dart-ui/Rect-class.html)? centerSlice, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) matchTextDirection = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) gaplessPlayback = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) isAntiAlias = false, [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html) filterQuality = FilterQuality.low})

Creates a widget that displays an image.

const

[Image.asset](https://api.flutter.dev/flutter/widgets/Image/Image.asset.html)([String](https://api.flutter.dev/flutter/dart-core/String-class.html) name, {[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [AssetBundle](https://api.flutter.dev/flutter/services/AssetBundle-class.html)? bundle, [ImageFrameBuilder](https://api.flutter.dev/flutter/widgets/ImageFrameBuilder.html)? frameBuilder, [ImageErrorWidgetBuilder](https://api.flutter.dev/flutter/widgets/ImageErrorWidgetBuilder.html)? errorBuilder, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticLabel, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeFromSemantics = false, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? scale, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? width, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? height, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? color, [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>? opacity, [BlendMode](https://api.flutter.dev/flutter/dart-ui/BlendMode.html)? colorBlendMode, [BoxFit](https://api.flutter.dev/flutter/painting/BoxFit.html)? fit, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment = Alignment.center, [ImageRepeat](https://api.flutter.dev/flutter/painting/ImageRepeat.html) repeat = ImageRepeat.noRepeat, [Rect](https://api.flutter.dev/flutter/dart-ui/Rect-class.html)? centerSlice, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) matchTextDirection = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) gaplessPlayback = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) isAntiAlias = false, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? package, [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html) filterQuality = FilterQuality.low, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheWidth, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheHeight})

Creates a widget that displays an [ImageStream](https://api.flutter.dev/flutter/painting/ImageStream-class.html) obtained from an asset bundle. The key for the image is given by the name argument.

[Image.file](https://api.flutter.dev/flutter/widgets/Image/Image.file.html)([File](https://api.flutter.dev/flutter/dart-io/File-class.html) file, {[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [double](https://api.flutter.dev/flutter/dart-core/double-class.html) scale = 1.0, [ImageFrameBuilder](https://api.flutter.dev/flutter/widgets/ImageFrameBuilder.html)? frameBuilder, [ImageErrorWidgetBuilder](https://api.flutter.dev/flutter/widgets/ImageErrorWidgetBuilder.html)? errorBuilder, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticLabel, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeFromSemantics = false, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? width, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? height, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? color, [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>? opacity, [BlendMode](https://api.flutter.dev/flutter/dart-ui/BlendMode.html)? colorBlendMode, [BoxFit](https://api.flutter.dev/flutter/painting/BoxFit.html)? fit, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment = Alignment.center, [ImageRepeat](https://api.flutter.dev/flutter/painting/ImageRepeat.html) repeat = ImageRepeat.noRepeat, [Rect](https://api.flutter.dev/flutter/dart-ui/Rect-class.html)? centerSlice, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) matchTextDirection = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) gaplessPlayback = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) isAntiAlias = false, [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html) filterQuality = FilterQuality.low, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheWidth, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheHeight})

Creates a widget that displays an [ImageStream](https://api.flutter.dev/flutter/painting/ImageStream-class.html) obtained from a [File](https://api.flutter.dev/flutter/dart-io/File-class.html).

[Image.memory](https://api.flutter.dev/flutter/widgets/Image/Image.memory.html)([Uint8List](https://api.flutter.dev/flutter/dart-typed_data/Uint8List-class.html) bytes, {[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [double](https://api.flutter.dev/flutter/dart-core/double-class.html) scale = 1.0, [ImageFrameBuilder](https://api.flutter.dev/flutter/widgets/ImageFrameBuilder.html)? frameBuilder, [ImageErrorWidgetBuilder](https://api.flutter.dev/flutter/widgets/ImageErrorWidgetBuilder.html)? errorBuilder, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticLabel, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeFromSemantics = false, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? width, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? height, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? color, [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>? opacity, [BlendMode](https://api.flutter.dev/flutter/dart-ui/BlendMode.html)? colorBlendMode, [BoxFit](https://api.flutter.dev/flutter/painting/BoxFit.html)? fit, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment = Alignment.center, [ImageRepeat](https://api.flutter.dev/flutter/painting/ImageRepeat.html) repeat = ImageRepeat.noRepeat, [Rect](https://api.flutter.dev/flutter/dart-ui/Rect-class.html)? centerSlice, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) matchTextDirection = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) gaplessPlayback = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) isAntiAlias = false, [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html) filterQuality = FilterQuality.low, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheWidth, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheHeight})

Creates a widget that displays an [ImageStream](https://api.flutter.dev/flutter/painting/ImageStream-class.html) obtained from a [Uint8List](https://api.flutter.dev/flutter/dart-typed_data/Uint8List-class.html).

[Image.network](https://api.flutter.dev/flutter/widgets/Image/Image.network.html)([String](https://api.flutter.dev/flutter/dart-core/String-class.html) src, {[Key](https://api.flutter.dev/flutter/foundation/Key-class.html)? key, [double](https://api.flutter.dev/flutter/dart-core/double-class.html) scale = 1.0, [ImageFrameBuilder](https://api.flutter.dev/flutter/widgets/ImageFrameBuilder.html)? frameBuilder, [ImageLoadingBuilder](https://api.flutter.dev/flutter/widgets/ImageLoadingBuilder.html)? loadingBuilder, [ImageErrorWidgetBuilder](https://api.flutter.dev/flutter/widgets/ImageErrorWidgetBuilder.html)? errorBuilder, [String](https://api.flutter.dev/flutter/dart-core/String-class.html)? semanticLabel, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) excludeFromSemantics = false, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? width, [double](https://api.flutter.dev/flutter/dart-core/double-class.html)? height, [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)? color, [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>? opacity, [BlendMode](https://api.flutter.dev/flutter/dart-ui/BlendMode.html)? colorBlendMode, [BoxFit](https://api.flutter.dev/flutter/painting/BoxFit.html)? fit, [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html) alignment = Alignment.center, [ImageRepeat](https://api.flutter.dev/flutter/painting/ImageRepeat.html) repeat = ImageRepeat.noRepeat, [Rect](https://api.flutter.dev/flutter/dart-ui/Rect-class.html)? centerSlice, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) matchTextDirection = false, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) gaplessPlayback = false, [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html) filterQuality = FilterQuality.low, [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html) isAntiAlias = false, [Map](https://api.flutter.dev/flutter/dart-core/Map-class.html)<[String](https://api.flutter.dev/flutter/dart-core/String-class.html), [String](https://api.flutter.dev/flutter/dart-core/String-class.html)>? headers, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheWidth, [int](https://api.flutter.dev/flutter/dart-core/int-class.html)? cacheHeight})

Creates a widget that displays an [ImageStream](https://api.flutter.dev/flutter/painting/ImageStream-class.html) obtained from the network.

### Properties

[alignment](https://api.flutter.dev/flutter/widgets/Image/alignment.html) → [AlignmentGeometry](https://api.flutter.dev/flutter/painting/AlignmentGeometry-class.html)

How to align the image within its bounds.

final

[centerSlice](https://api.flutter.dev/flutter/widgets/Image/centerSlice.html) → [Rect](https://api.flutter.dev/flutter/dart-ui/Rect-class.html)?

The center slice for a nine-patch image.

final

[color](https://api.flutter.dev/flutter/widgets/Image/color.html) → [Color](https://api.flutter.dev/flutter/dart-ui/Color-class.html)?

If non-null, this color is blended with each image pixel using [colorBlendMode](https://api.flutter.dev/flutter/widgets/Image/colorBlendMode.html).

final

[colorBlendMode](https://api.flutter.dev/flutter/widgets/Image/colorBlendMode.html) → [BlendMode](https://api.flutter.dev/flutter/dart-ui/BlendMode.html)?

Used to combine [color](https://api.flutter.dev/flutter/widgets/Image/color.html) with this image.

final

[errorBuilder](https://api.flutter.dev/flutter/widgets/Image/errorBuilder.html) → [ImageErrorWidgetBuilder](https://api.flutter.dev/flutter/widgets/ImageErrorWidgetBuilder.html)?

A builder function that is called if an error occurs during image loading.

final

[excludeFromSemantics](https://api.flutter.dev/flutter/widgets/Image/excludeFromSemantics.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether to exclude this image from semantics.

final

[filterQuality](https://api.flutter.dev/flutter/widgets/Image/filterQuality.html) → [FilterQuality](https://api.flutter.dev/flutter/dart-ui/FilterQuality.html)

The rendering quality of the image.

final

[fit](https://api.flutter.dev/flutter/widgets/Image/fit.html) → [BoxFit](https://api.flutter.dev/flutter/painting/BoxFit.html)?

How to inscribe the image into the space allocated during layout.

final

[frameBuilder](https://api.flutter.dev/flutter/widgets/Image/frameBuilder.html) → [ImageFrameBuilder](https://api.flutter.dev/flutter/widgets/ImageFrameBuilder.html)?

A builder function responsible for creating the widget that represents this image.

final

[gaplessPlayback](https://api.flutter.dev/flutter/widgets/Image/gaplessPlayback.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether to continue showing the old image (true), or briefly show nothing (false), when the image provider changes. The default value is false.

final

[hashCode](https://api.flutter.dev/flutter/widgets/Widget/hashCode.html) → [int](https://api.flutter.dev/flutter/dart-core/int-class.html)

The hash code for this object.

read-onlyinherited

[height](https://api.flutter.dev/flutter/widgets/Image/height.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

If non-null, require the image to have this height (in logical pixels).

final

[image](https://api.flutter.dev/flutter/widgets/Image/image.html) → [ImageProvider](https://api.flutter.dev/flutter/painting/ImageProvider-class.html)<[Object](https://api.flutter.dev/flutter/dart-core/Object-class.html)>

The image to display.

final

[isAntiAlias](https://api.flutter.dev/flutter/widgets/Image/isAntiAlias.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether to paint the image with anti-aliasing.

final

[key](https://api.flutter.dev/flutter/widgets/Widget/key.html) → [Key](https://api.flutter.dev/flutter/foundation/Key-class.html)?

Controls how one widget replaces another widget in the tree.

finalinherited

[loadingBuilder](https://api.flutter.dev/flutter/widgets/Image/loadingBuilder.html) → [ImageLoadingBuilder](https://api.flutter.dev/flutter/widgets/ImageLoadingBuilder.html)?

A builder that specifies the widget to display to the user while an image is still loading.

final

[matchTextDirection](https://api.flutter.dev/flutter/widgets/Image/matchTextDirection.html) → [bool](https://api.flutter.dev/flutter/dart-core/bool-class.html)

Whether to paint the image in the direction of the [TextDirection](https://api.flutter.dev/flutter/dart-ui/TextDirection.html).

final

[opacity](https://api.flutter.dev/flutter/widgets/Image/opacity.html) → [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html)<[double](https://api.flutter.dev/flutter/dart-core/double-class.html)>?

If non-null, the value from the [Animation](https://api.flutter.dev/flutter/animation/Animation-class.html) is multiplied with the opacity of each image pixel before painting onto the canvas.

final

[repeat](https://api.flutter.dev/flutter/widgets/Image/repeat.html) → [ImageRepeat](https://api.flutter.dev/flutter/painting/ImageRepeat.html)

How to paint any portions of the layout bounds not covered by the image.

final

[runtimeType](https://api.flutter.dev/flutter/dart-core/Object/runtimeType.html) → [Type](https://api.flutter.dev/flutter/dart-core/Type-class.html)

A representation of the runtime type of the object.

read-onlyinherited

[semanticLabel](https://api.flutter.dev/flutter/widgets/Image/semanticLabel.html) → [String](https://api.flutter.dev/flutter/dart-core/String-class.html)?

A Semantic description of the image.

final

[width](https://api.flutter.dev/flutter/widgets/Image/width.html) → [double](https://api.flutter.dev/flutter/dart-core/double-class.html)?

If non-null, require the image to have this width (in logical pixels).

## raw image

A widget that displays a [dart:ui.Image](https://api.flutter.dev/flutter/dart-ui/Image-class.html) directly.

The image is painted using [paintImage](https://api.flutter.dev/flutter/painting/paintImage.html), which describes the meanings of the various fields on this class in more detail.

The [image](https://api.flutter.dev/flutter/widgets/RawImage/image.html) is not disposed of by this widget. Creators of the widget are expected to call [Image.dispose](https://api.flutter.dev/flutter/dart-ui/Image/dispose.html) on the [image](https://api.flutter.dev/flutter/widgets/RawImage/image.html) once the [RawImage](https://api.flutter.dev/flutter/widgets/RawImage-class.html) is no longer buildable.

This widget is rarely used directly. Instead, consider using [Image](https://api.flutter.dev/flutter/widgets/Image-class.html).