



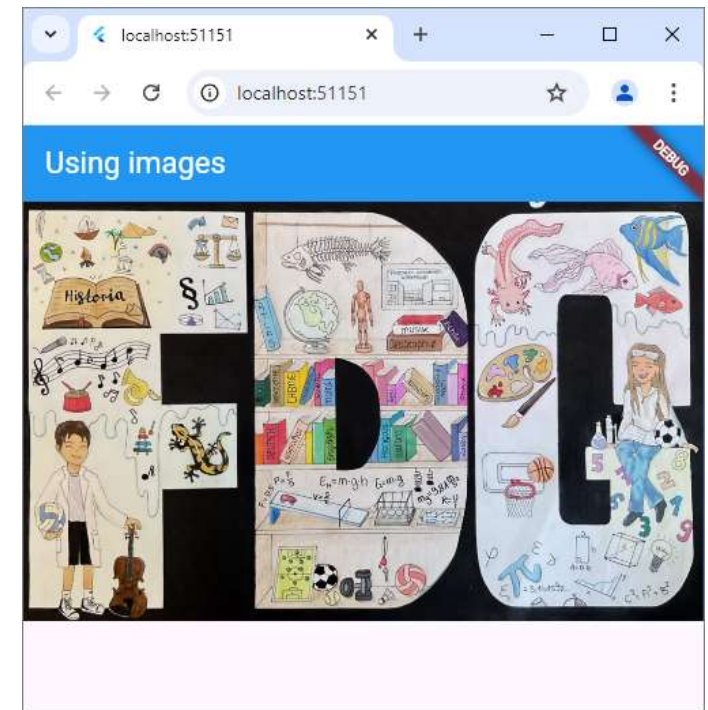
Use images in your app

- Show images from the internet in your app
- Learn how to size images
- Enable vertical scrolling with widget SingleChildScrollView
- Know how to use Padding
- Use images that are bundled inside your app
- Know that png images can have a transparent background



Show Images from the network

```
Widget build(BuildContext context) {  
  return MaterialApp(  
    home: Scaffold(  
      appBar: AppBar(  
        title: Text("Using images"),  
        backgroundColor: Colors.blue,  
        foregroundColor: Colors.white,  
      ), // AppBar  
      body: Image.network(  
        "https://fdg2324.github.io/web/fdg_leitbild.png",  
      ), // Image.network  
    ), // Scaffold  
  ); // MaterialApp  
}
```



Link to the used FDG image for copy/paste: https://fdg2324.github.io/web/fdg_leitbild.png

Changing the size of the browser window

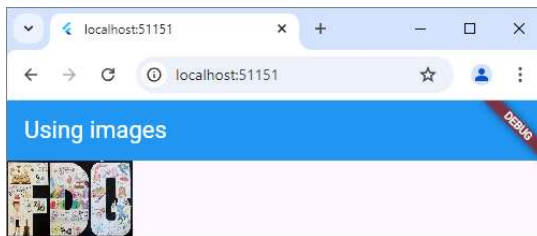


Image shrinks if needed and keeps its aspect ratio (Seitenverhältnis).

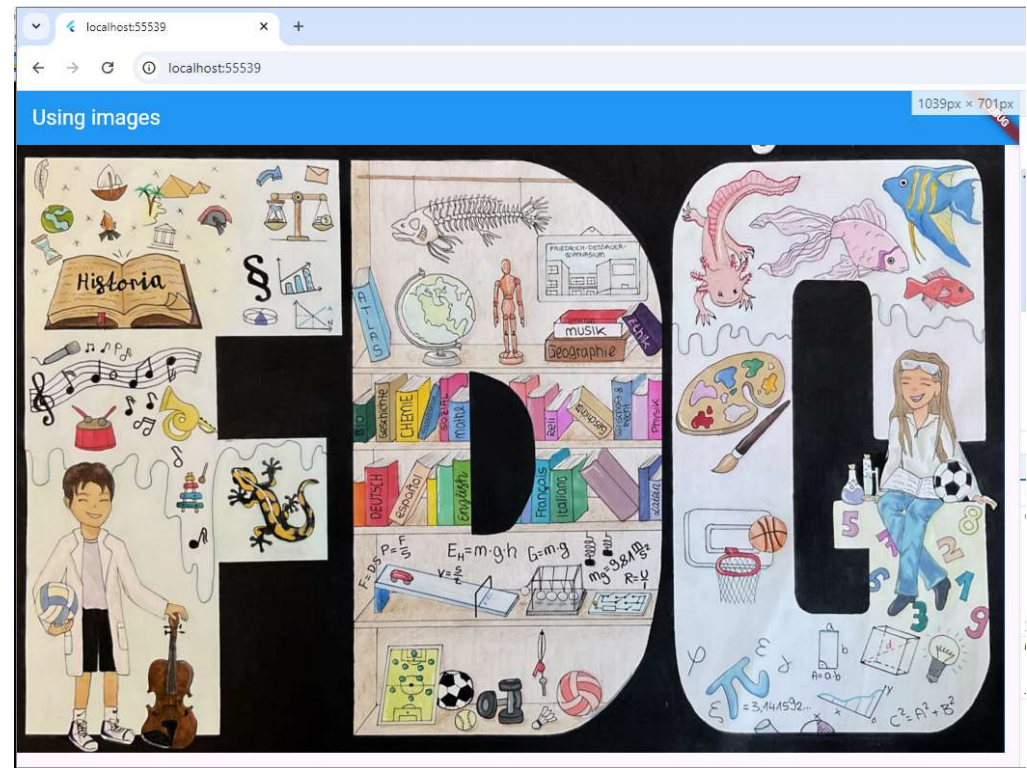


Image does not grow beyond its own size (here 1024 * 630 Pixel).



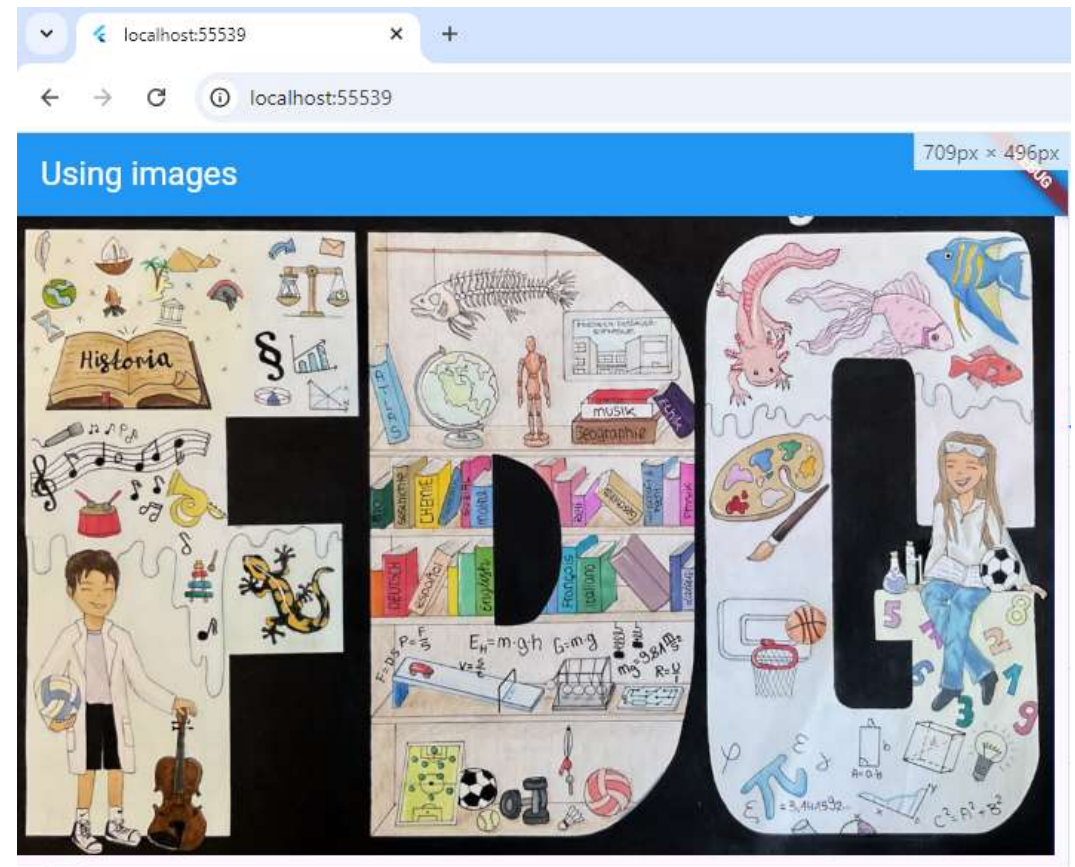
Setting width or height of an image

```
body: Image.network(  
  width: 700,  
  //height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
) , // Image.network
```

Setting the width of an image restricts its maximum width.

The image shrinks when the windows size gets smaller.

Same for the height.

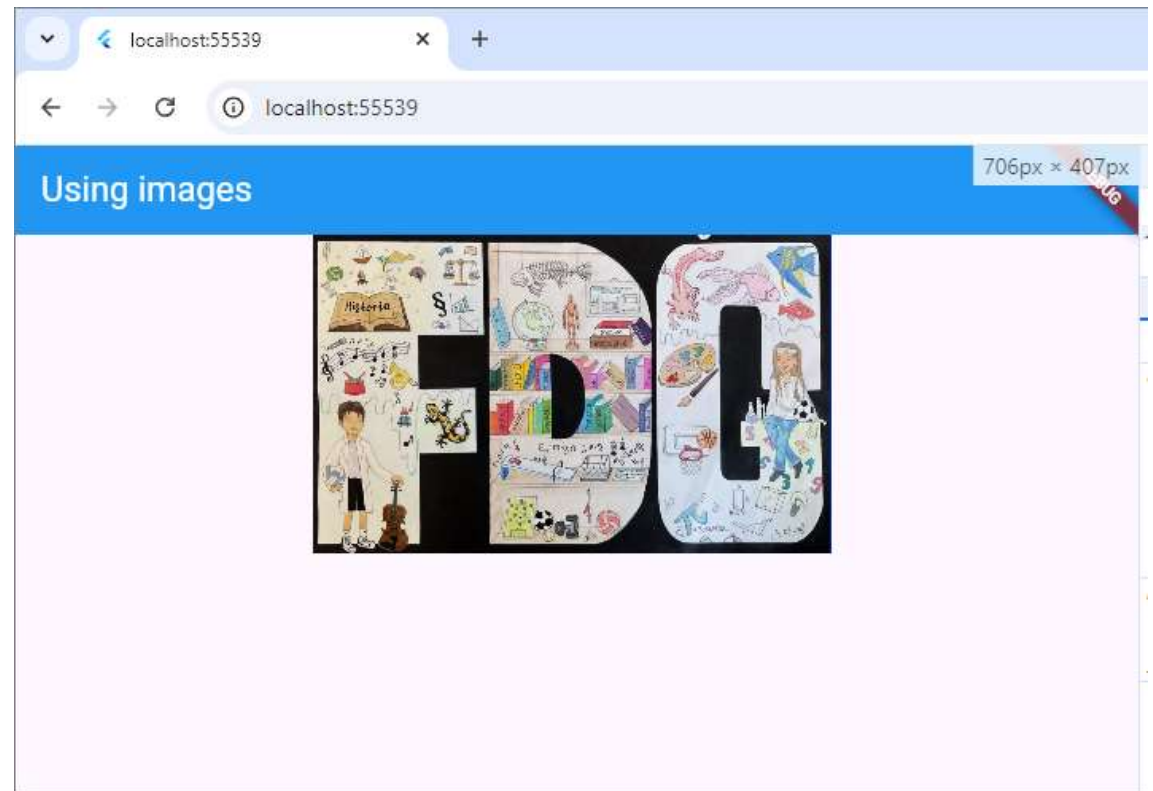




Setting width and height of an image

```
body: Image.network(  
  width: 700,  
  height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
) // Image.network
```

When both width and height are set, the “most restrictive wins”.



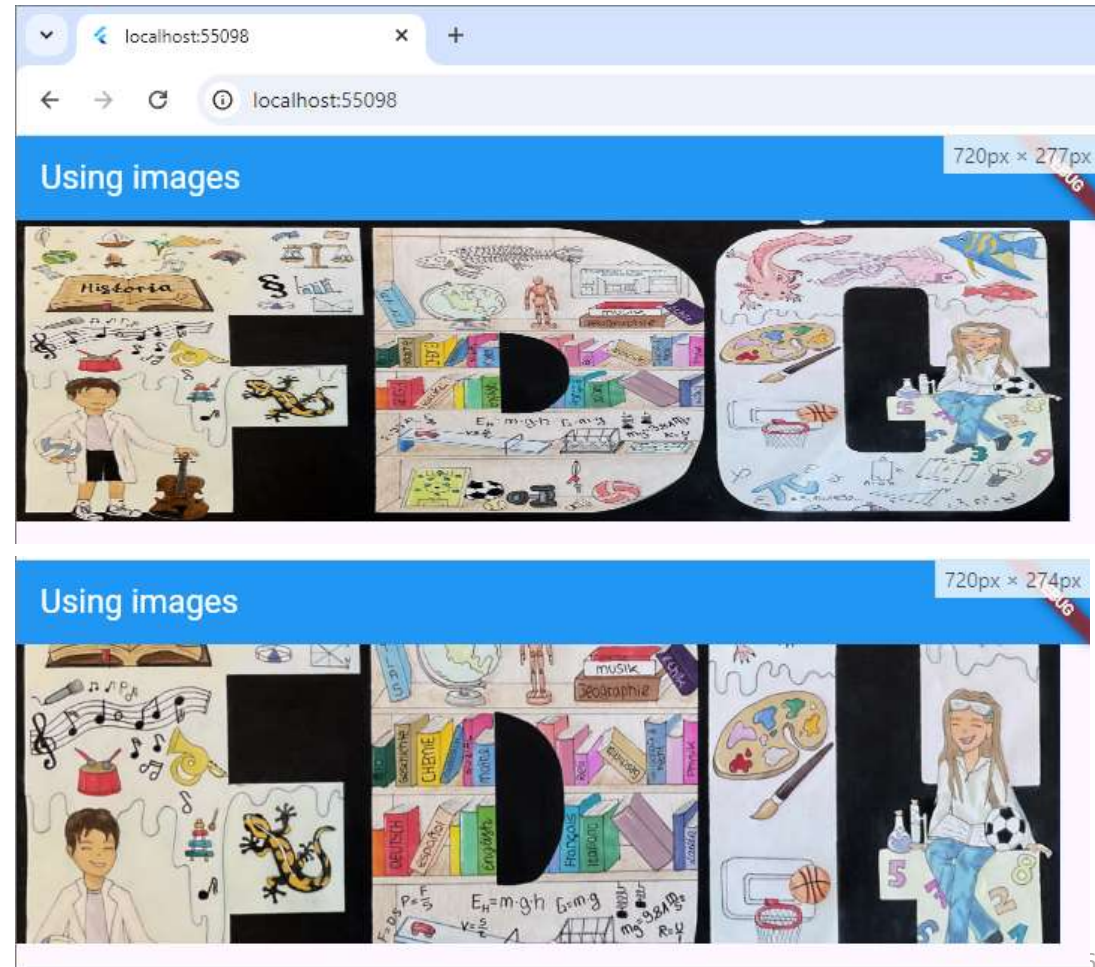


Boxfit for images

```
body: Image.network(  
  width: 700,  
  height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
  fit: BoxFit.fill  
) , // Image.network
```

Aspect ratio distorted !

```
body: Image.network(  
  width: 700,  
  height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
  fit: BoxFit.cover  
) , // Image.network
```





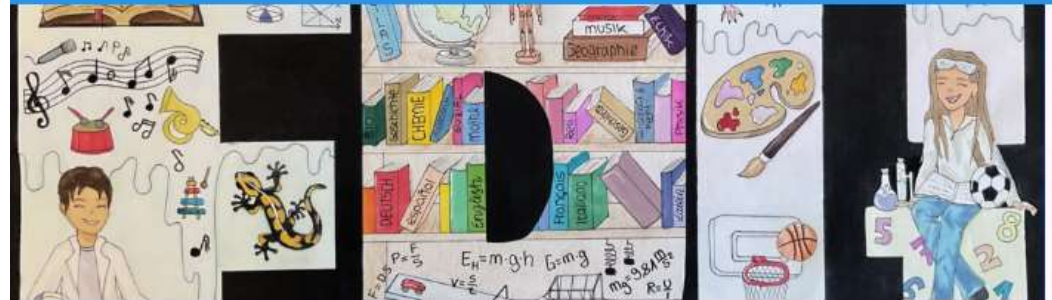
Boxfit for images (continued)

```
body: Image.network(  
  width: 700,  
  height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
  fit: BoxFit.fitWidth  
) , // Image.network
```

```
body: Image.network(  
  width: 700,  
  height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
  fit: BoxFit.fitHeight  
) , // Image.network
```

Using images

724px × 276px



Using images

724px × 268px





Boxfit for images (continued)

```
body: Image.network(  
  width: 700,  
  height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
  fit: BoxFit.contain  
)
```

Using images

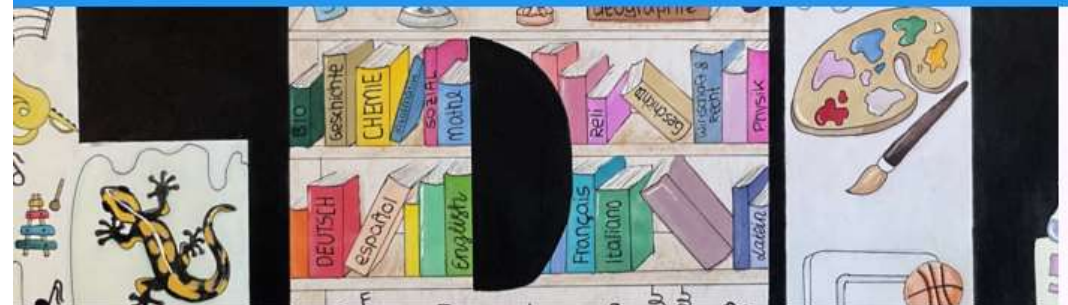
726px × 282px



```
body: Image.network(  
  width: 700,  
  height: 200,  
  "https://fdg2324.github.io/web/fdg_leitbild.png",  
  fit: BoxFit.none,  
)
```

Using images

726px × 278px



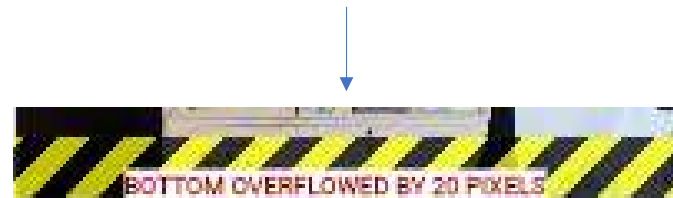
For more info see <https://api.flutter.dev/flutter/painting/BoxFit.html>



Image inside a Column

```
body: Column(  
  children: [  
    Image.network(  
      width: 700,  
      height: 200,  
      "https://fdg2324.github.io/web/fdg_leitbild.png",  
    ), // Image.network  
  ],  
) // Column
```

The height of the image no longer shrinks when it is inside a Column.

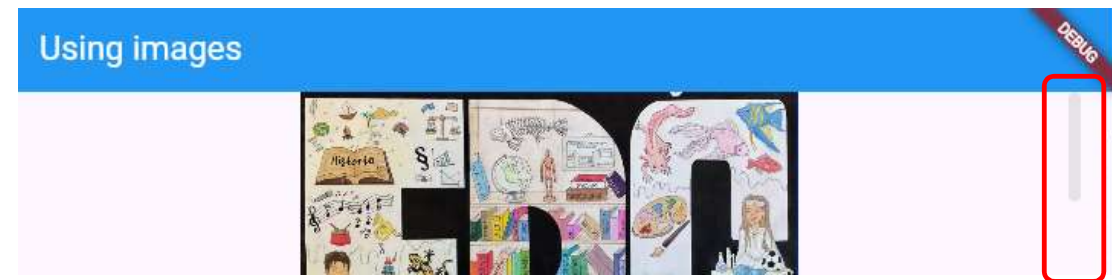




SingleChildScrollView

```
body: SingleChildScrollView(  
  child: Column(  
    children: [  
      Image.network(  
        width: 700,  
        height: 200,  
        "https://fdg2324.github.io/web/fdg_leitbild.png",  
      ), // Image.network  
    ],  
  ), // Column  
) // SingleChildScrollView
```

Wrapping the Column inside a
SingleChildScrollView widget
let's the user scroll up and down.





Compared to Chrome, Android phones have a fixed screen size

```
body: SingleChildScrollView(  
  child: Column(  
    children: [  
      Image.network(  
        width: 700,  
        //height: 200,  
        "https://fdg2324.github.io/web/fdg_leitbild.png",  
      ), // Image.network  
      Image.network(  
        width: 390,  
        //height: 200,  
        "https://fdg2324.github.io/web/fdg_leitbild.png",  
      ), // Image.network  
    ],  
  ), // Column  
) // SingleChildScrollView
```



“Virtual Device Manager”
in Android Studio:

Medium Phone API 35	
Device Info	Paired Devices
Summary	
API level	35
Resolution (px)	1080 × 2400
Resolution (dp)	412 × 915
Density	420 dpi
ABI list	x86_64
Available storage	2,192 MB
Size on disk	9,7 GB





Different Android phones can have different screen sizes



Medium Phone API 35	
Device Info	Paired Devices
Summary	
API level	35
Resolution (px)	1080 × 2400
Resolution (dp)	412 × 915
Density	420 dpi
ABI list	x86_64
Available storage	2,192 MB
Size on disk	9,7 GB



Pixel 3a API 35	
Device Info	Paired Devices
Summary	
API level	35
Resolution (px)	1080 × 2220
Resolution (dp)	393 × 808
Density	440 dpi
ABI list	x86_64
Available storage	5,1 GB
Size on disk	5,1 GB



Padding widget

```
body: SingleChildScrollView(  
  child: Column(  
    children: [  
      Image.network(  
        width: 700,  
        //height: 200,  
        "https://fdg2324.github.io/web/fdg_leitbild.png",  
      ), // Image.network  
      Padding(  
        padding: const EdgeInsets.all(20.0),  
        child: Image.network(  
          width: 700,  
          //height: 200,  
          "https://fdg2324.github.io/web/fdg_leitbild.png",  
        ), // Image.network  
      ), // Padding  
    ],  
  ), // Column  
), // SingleChildScrollView
```



Using EdgeInsets.fromLTRB

EdgeInsets has several constructors:

```
EdgeInsets EdgeInsets.fromLTRB(  
  double left,  
  double top,  
  double right,  
  double bottom,  
)  
  
package:flutter/src/painting/edge_insets.dart  
  
Creates insets from offsets from the left, top, right, and bottom.
```

```
Padding(  
  padding: const EdgeInsets.all(20.0),  
  child: Image.network(  
    width: 700,  
    //height: 200,  
    "https://fdg2324.github.io/web/fdg_leitbild.png",  
  ), // Image.network  
) // Padding
```



```
Padding(  
  padding: const EdgeInsets.fromLTRB(10.0, 20, 30, 0),  
  child: Image.network(  
    width: 700,  
    //height: 200,  
    "https://fdg2324.github.io/web/fdg_leitbild.png",  
  ), // Image.network  
) // Padding
```





Exercise

Look for an image of the FDG logo in the internet:



Use the link to this image to display it in your Android app.

Define a padding for this image and use constructor “`EdgeInsets.symmetric`” inside the `Padding`.

Change the background color of your app’s body e.g. to “`lightBlue.shade200`” (hint: look for corresponding property in `Scaffold`)



Solution

The link to the image is:

https://fdg-ab.de/wp-content/uploads/2021/03/logo_fdg_neu_freigestellt.png

Image with padding:

```
Padding(  
  padding: const EdgeInsets.symmetric(vertical: 20, horizontal: 50),  
  child: Image.network("https://fdg-ab.de/wp-content/uploads/2021/03/logo_fdg_neu_freigestellt.png"),  
), // Padding
```

Background of the app:

```
Widget build(BuildContext context) {  
  return MaterialApp(  
    home: Scaffold(  
      backgroundColor: Colors.lightBlue.shade200,  
      appBar: AppBar(  

```



Remark:

Transparent backgrounds can only be defined in **png** images, not in **jpg** images !



Some image URL's do not work in Chrome

The original link to the image used in the first slides is

<https://fdg-ab.de/wp-content/uploads/2022/08/Bildschirmfoto-2022-08-10-um-13.07.47-1024x630.png>

This works in Android, but not when debugging in Chrome:

```
body: SingleChildScrollView(  
  child: Column(  
    children: [  
      Image.network(  
        "https://fdg2324.github.io/web/fdg_leitbild.png",  
      ), // Image.network  
      Image.network(  
        "https://fdg-ab.de/wp-content/uploads/2022/08/Bildschirmfoto-2022-08-10-um-13.07.47-1024x630.png",  
      ), // Image.network  
    ],  
  ), // Column  
), // SingleChildScrollView
```

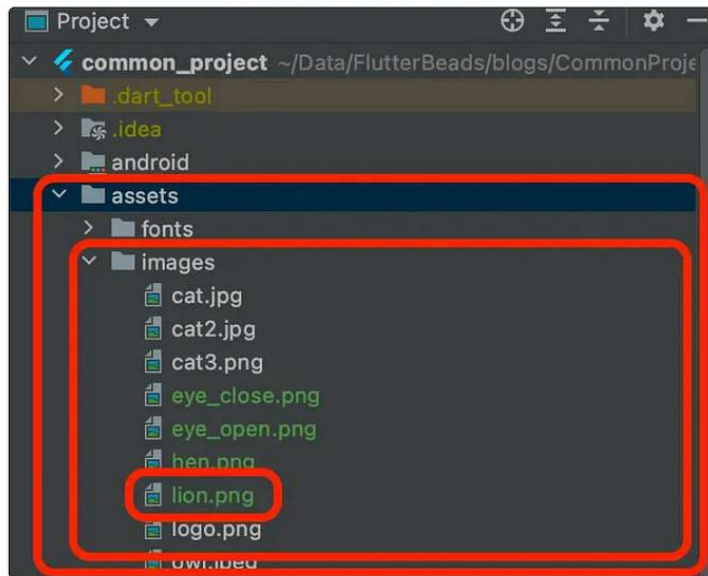




Bundle images inside your app

This is good for Android too in case the phone has no internet connection.

Copied from <https://medium.com/@blup-tool/learn-flutter-how-to-add-images-in-your-flutter-app-359c27600064> :



```
Column(  
  children: [  
    Image.asset('assets/images/lion.png'),  
  ],  
)
```

Take care:

use forward slashes “/” as separators,
no “backward slashes” \ as in Windows!

Having a subdirectory **assets** is recommended, but not needed.

You can have an “images” or “pictures” directory directly inside your project.



Change needed in pubspec.yaml file

```
! pubspec.yaml
1  name: using_images
2  description: "A new Flutter project."
3  publish_to: 'none'
4  version: 0.1.0
5
6  environment:
7    sdk: ^3.5.3
8
9  dependencies:
10    flutter:
11      sdk: flutter
12
13  dev_dependencies:
14    flutter_test:
15      sdk: flutter
16    flutter_lints: ^4.0.0
17
18  flutter:
19    uses-material-design: true
20
21    assets:
22      - assets/images/
```

YAML

[Article](#) [Talk](#)

From Wikipedia, the free encyclopedia

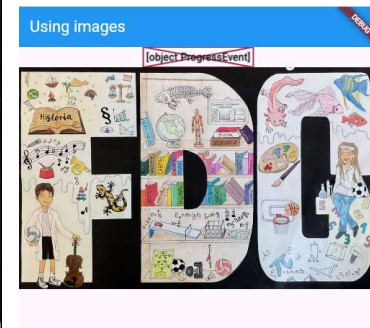
For the CSS framework, see [YAML \(framework\)](#).

YAML (ⁱˈjæməl [ⓘ]; see § [History and name](#)) is a [human-readable data serialization language](#). It is commonly used for [configuration files](#) and in applications where data is being stored or transmitted. YAML targets many of the same communications applications as [Extensible Markup Language](#) (XML) but has a minimal syntax that intentionally differs from [Standard Generalized Markup Language](#) (SGML).^[3] It **uses Python-style indentation to indicate nesting**^[3] and does not require quotes around most string values (it also supports [JSON](#) style `[...]` and `{...}` mixed in the same file).^[4]



Images from asset work both in Chrome and Android

```
body: SingleChildScrollView(  
  child: Column(  
    children: [  
      Image.network(  
        "https://fdg-ab.de/wp-content/uploads/2022/08/Bildschirmfoto-2022-08-10-um-13.07.47-1024x630.png"),  
      Image.asset(  
        "assets/images/Bildschirmfoto-2022-08-10-um-13.07.47-1024x630.png") // Image.asset  
    ],  
  ), // Column  
), // SingleChildScrollView
```





Exercise

Download the FDG logo image to your PC.

Bundle it into your app.

Display it in the leading part of the AppBar
(this was our goal in “04 Scaffold and AppBar.pptx”):



Hint: use a property of `image.asset` to show the logo in white.

Solution:

```
appBar: AppBar(  
  title: Text("Flutter at FDG"),  
  centerTitle: true,  
  backgroundColor: Colors.blue,  
  foregroundColor: Colors.white,  
  leading: Padding(  
    padding: const EdgeInsets.all(10),  
    child: Image.asset("assets/images/fdg_logo.png", color: Colors.white),  
  ), // Padding  
), // AppBar
```