

Asset 资源管理
Asset 变体
名词解释:
用途:
加载文本 assets
加载图片

Asset 资源管理

我们可以在 配置文件 pubspec.yaml文件中，指定flutter的 asset 文件。

```
1 flutter:
2   assets:
3     - assets/my_icon.png
4     - assets/background.png
```

Asset 变体

名词解释:

如果在我们指定的资源文件夹 /A/下，定义了资源/A/a.png。
在 /A 下的 子文件夹 /A/B/ 中，我们定义了相同名字的资源文件 /A/B/a.png。
那么， /A/B/a.png 称作是 /A/a.png 的变体。
父文件夹中的 /A/a.png 被称为主资源。

用途:

匹配设备分辨率的时候，会用到asset变体。

加载文本 assets

两种方式加载assets:

1. rootBundle. 每个Flutter 应用都对应一个rootBundle对象。通过该对象能够轻松访问竹资源包。

2. DefaultAssetBundle.

- a. 当我们处于 widget上下文环境中时, 通过这种方式是建议使用的方法, 来获取asset资源。
- b. 如果我们处于widget之外, 或者其他AssetBundle句柄不可用时, 可以使用rootBundle直接加载asset。

加载图片

AssetImage 用于将 asset的请求逻辑 映射到最接近 当前设备像素比例(dpi) 的asset上。
类似于Android, 我们必须采用对应的目录结构来存储资源。

```
1  .... /image.png
2  ... /Mx/image.png
3  ... /Nx/image.png
4  .....
```

其中 M 和 N 为数字, eg: 2.0 3.0, 主资源默认对应于 1.0x的路径, 即前面没有 M N 路径的。
当主资源项缺少某个资源时, 会按照分辨率从低到高的顺序, 去对应路径下查找资源。

注意:

- 1 使用 asset bundle 加载资源, 会默认帮我们处理资源的分辨率等,
- 2 但是如果使用 ImageStream 或 ImageCache 可能需要我们处理缩放相关的参数。

加载资源时, 需要指定package包名。

```
1 new AssetImage('icons/heart.png',package: 'my_icons')
```

打包包中的assets

我们必须在pubspec.yaml 文件中指定图片。 如果 我们想引入没有在 pubspec.yaml 中定义的, 而只是在lib/路径下定义的图片,
那么我们是需要在 pubsepc.yaml中引入他的, 同时在指定的时候, 是不需要显式地加入 lib/引用路径的。

eg 我们在一个包名为 fancy 的包下, 有以下的图片:

```
1  ... /lib/background/backgrounds.png
```

引入assets 代码:

```
1 flutter:
2   assets:
```

```
3      - packages/fancy/backgrounds.png.  
4      # 这里的fancy是包名，省略了lib 路径。
```

原因是 lib/是他的资产路径。