

1. 使用 AssetServer 从文件加载资产

要从文件加载资产，请使用 AssetServer 资源。

```
fn setup(mut commands: Commands, asset_server:
Res<AssetServer>) {
    commands.spawn(Camera2dBundle::default());

    commands.spawn(SpriteBundle {
        texture: asset_server.load("14_load_assets/
bevy_bird_dark.png"),
        ..default()
    });
}
```

这会将资产加载排队到后台，并返回一个句柄。资产需要一些时间才能变得可用。您无法在同一系统中立即访问实际数据，但可以使用句柄。

您可以使用句柄生成实体，如 2D 精灵、3D 模型和 UI，即使在资产加载之前也是如此。它们将在资产准备好后“弹出”。

请注意，即使资产当前正在加载或已经加载，您也可以随时调用 `asset_server.load(...)`。它只会为您提供相同的句柄。每次调用时，它只会检查资产的状态，如果需要，开始加载并给您一个句柄。

Bevy 支持加载各种资产文件格式，并且可以扩展以支持更多格式。要使用的资产加载器实现是根据文件扩展名选择的。

2. 资产路径和标签

用于从文件系统中识别资产的资产路径实际上是一个特殊的 `AssetPath`，它由文件路径和标签组成。标签用于在同一个文件中包含多个资产的情况。例如，GLTF 文件可以包含网格、场景、纹理、材质等。

资产路径可以从字符串创建，标签（如果有）附加在 `#` 符号之后。

```
fn setup(
    mut commands: Commands,
    asset_server: Res<AssetServer>,
) {
    commands.spawn(SceneBundle {
        // asset_server.load("14_load_assets/
grass.glb#Scene0");
        scene:
asset_server.load(GltfAssetLabel::Scene(0).from_asset("14_load_assets/
grass.glb")),
        ..default()
    });
}
```