<https://blog.csdn.net/ldghd/article/details/8069489>

Unity里面有两种动态加载机制：一是Resources.Load，一是通过AssetBundle。其实两者本质上没有什么区别，Resources.Load就是从一个缺省打进程序包里的AssetBundle里加载资源，而一般AssetBundle文件需要你自己创建，运行时动态加载，可以指定路径和来源，其实场景中所有静态的对象也有这么一个加载过程，只是Unity后台自动完成了。

**AssetBundle运行时加载**

来自文件就用CreateFromFile（注意这种方法只能用于standalone程序），这是最快的加载方法

也可以来自Memory，用CreateFromMemory(byte[])，这个byte[]可以来自文件读取的缓冲，www的下载或者其他可能的方式

其实www的assetBundle就是内部数据读取万后自动创建了一个assetBundle而已

Create完成以后，等于把硬盘或者网络的一个文件读到内存的 一个区域，这时候只是一个AssetBundle内存镜像数据块，还没有Assets的概念

Assets加载：

用AssetBundle.Load（通Resources.Load）这才会从AssetBundle的内存镜像中读取并创建一个Asset对象，创建Asset对象同时也会分配相应内存用于存放（反序列化），异步读取用AssetBundle.LoadAsync，也可以一次读取多个，用AssetBundle.LoadAll

**AssetBundle的释放**

AssetBundle.Unload(false)是释放AssetBundle文件的内存镜像，不包含Load创建的Asset内存对象

AssetBundle.Unload(true)是释放AssetBundle文件的内存镜像并销毁所有用 Load创建的Asset内存对象