<https://blog.csdn.net/u010377179/article/details/52922710>

<https://blog.csdn.net/Daniel_love/article/details/51699128>

<http://www.cnblogs.com/ybgame/p/3973177.html>

<https://blog.csdn.net/langresser_king/article/details/44208585/>

<https://blog.csdn.net/huutu/article/details/46477471>

<https://blog.csdn.net/hanghangaidoudou/article/details/78170591>

<https://www.cnblogs.com/lan-yt/p/7787290.html>

<https://www.cnblogs.com/WongSiuming/p/7574312.html>

<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。

**采用Resources加载资源**

GameObject prefab = (GameObject)Resources.Load("prefab2", typeof(GameObject));

GameObject t = Instantiate(prefab) as GameObject;

t.name = "clone\_prefab1";

Debug.Log("name = " + prefab.name);

在上面的例子中尝试过建立多个Resources文件，实际上都可以找到prefab，注意Resources.Load不需要添加资源后缀名

**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

**什么是AssetBundle？**

AssetBundle只是一种使用LZMA压缩方式压缩的资源文件，至于它的后缀名是什么，一点关系都没有，可以自己定。

AssetBundle打包的时，可以指定一个mainAsset，那么加载完之后就可以通过AssetBundle.mainAsset来获取到了，也可以不指定mainAsset，直接打包一堆内容进去，然后加载通过AssetBundle.LoadAsset指定名字读取出来。在资源之间，存在着依赖关系，你可以把资源拆分得很细，比如一个模型，由网格模型、材质、贴图构成，可以将他们分开，各自打包成压缩文件。当Unity需要加载使用时，把该模型的所有依赖的小资源都加载起来，然后根据依赖关系组装就变回我们看到的资源了。

**打包AssetBundle**

Unity5.x大幅度简化了AssetBundle的打包过程，理论上只需要调用一个API既可以完成打包, BuildPipeline.BuildAssetBundles

但是资源之间的依赖关系，避免资源重复打包的问题还是要自己解决

Unity5r.x中会将设置AssetBundleName相同的资源打包到一起，所以我们的打包过程其实就是把需要打包到一起的资源设置成相同的AssetBundleName，为每个资源设置AssetBundleName，最后调用打包API即可完成打包

**AssetBundle的依赖结构**

要说明依赖关系，还需要使用一个例子，一个模型，分为了网络模型、材质、贴图。那么它们是怎样依赖的呢？然后在unity5的打包里面，它们是怎样表现出依赖关系的呢？