在Unity3D脚本中，有几个Unity3D自带的事件函数按照预定的顺序执行作为脚本执行。其执行顺序如下：

**编辑器（Editor）**

* **Reset：**Reset函数被调用来初始化脚本属性当脚本第一次被附到对象上，并且在Reset命令被使用时也会调用。  
  编者注：Reset是在用户点击Inspector面板上Reset按钮或者首次添加该组件时被调用。Reset最常用于在见识面板中给定一个默认值。

**第一次场景加载（First Scene Load）**  
这些函数会在一个场景开始（场景中每个物体只调用一次）时被调用。

* **Awake：**这个函数总是在任何Start()函数之前一个预设被实例化之后被调用，如果一个GameObject是非激活的（inactive），在启动期间Awake函数是不会被调用的直到它是活动的（active）。
* **OnEnable：**只有在对象是激活（active）状态下才会被调用，这个函数只有在object被启用（enable）后才会调用。这会发生在一个MonoBehaviour实例被创建，例如当一个关卡被加载或者一个带有脚本组件的GameObject被实例化。

注意：当一个场景被添加到场景中，所有脚本上的Awake()和OnEable()函数将会被调用在Start()、Update()等它们中任何函数被调用之前。自然的，当一个物体在游戏过程中被实例化时这不能被强制执行。

**第一帧更新之前（Before the first frame update）**

* **Start:**只要脚本实例被启用了Start()函数将会在Update()函数第一帧之前被调用。

对于那些被添加到场景中的物体，所有脚本上的Start()函数将会在它们中任何的Update()函数之前被调用，自然的，当一个物体在游戏过程中被实例化时这不能被强制执行。

**在帧之间（In between frames）**

1. **OnApplicationPause：**这个函数将会被调用在暂停被检测有效的在正常的帧更新之间的一帧的结束时。在OnApplicationPause被调用后将会有额外的一帧用来允许游戏显示显示图像表示在暂停状态下。

**更新顺序（Update Order）**

当你在跟踪游戏逻辑和状态，动画，相机位置等的时候，有几个不同的事件函数你可以使用。常见的模式是在Update()函数中执行大多数任务，但是也有其它的函数你可以使用。

* **FixedUpdate：**FixedUpdate函数经常会比Update函数更频繁的被调用。它一帧会被调用多次，如果帧率低它可能不会在帧之间被调用，就算帧率是高的。所有的图形计算和更新在FixedUpdate之后会立即执行。当在FixedUpdate里执行移动计算，你并不需要Time.deltaTime乘以你的值，这是因为FixedUpdate是按真实时间，独立于帧率被调用的。
* **Update：**Update每一帧都会被调用，对于帧更新它是主要的负荷函数。
* **LateUpdate：**LateUpdate会在Update结束之后每一帧被调用，任何计算在Update里执行结束当LateUpdate开始时。LateUpdate常用为第三人称视角相机跟随。

**渲染（Rendering）**

1. **OnPreCull：**在相机剔除场景前被调用。剔除是取决于哪些物体对于摄像机是可见的，OnPreCull仅在剔除起作用之前被调用。
2. **OnBecameVisible/OnBecameInvisible：**当一个物体对任意摄像机变得可见/不可见时被调用。
3. **OnPreRender：**在摄像机开始渲染场景之前调用。
4. **OnRenderObject：**在指定场景渲染完成之后调用，你可以使用GL类或者Graphics.DrawMeshNow 来绘制自定义几何体在这里。
5. **OnPostRender：**在摄像机完成场景渲染之后调用。
6. **OnRenderImage(Pro Only)：**在场景徐然完成之后允许屏幕图像后期处理调用。
7. **OnGUI：**为了响应GUI事件，每帧会被调用多次（一般最低两次）。布局Layout和Repaint事件会首先处理，接下来处理的是是通过  
   Layout和键盘/鼠标事件对应的每个输入事件。
8. **OnDrawGizmos：**用于可视化的绘制一些小玩意在场景视图中。

**协同程序（Coroutines）**

正常的协同程序更新是在Update函数返回之后运行。一个协同程序是可以暂停执行（yield）直到给出的依从指令（YieldInstruction ）完成，写成的不同运用：

* **yield：**在所有的Update函数都已经被调用的下一帧该协程将持续执行。
* **yield WaitForSeconds：**一段指定的时间延迟之后继续执行，在所有的Update函数完成调用的那一帧之后。
* **yield WaitForFixedUpdate：**所有脚本上的FixedUpdate函数已经执行调用之后持续。
* **yield WWW：**在WWW下载完成之后持续。
* **yield StartCoroutine：**协同程序链，将会等到MuFunc函数协程执行完成首先。

**销毁（When the Object is Destroyed）**

* **OnDestory:**这个函数在会在一个对象销毁前一帧调用，会在所有帧更新一个对象存在的最后一帧之后执行，对象也许会响应Object.Destroy 或一个场景关闭时被销毁。

**退出游戏（When Quitting）**  
这些函数会在你场景中所有的激活的物体上调用：

1. **OnApplicationQuit：**这个函数在应用退出之前的所有游戏物体上调用，在编辑器（Editor）模式中会在用户停止PlayMode时调用，在网页播放器（web player）中会在网页视图关闭时调用。
2. **OnDisable：**当行为变为非启用（disable）或非激活（inactive）时调用。

**脚本的生命周期流程图**





