<http://www.unity.5helpyou.com/3550.html> （全面）

Unity会自动为public变量做序列化，序列化的意思是说再次读取unity时序列化的变量是有值的，不需要你再次去赋值，因为它已经被保存起来了。

**[HideInInspector]**表示将原本显示在面板上的值隐藏起来

**[SerializeField]**表示将原本不会被序列化的私有变量和保护变量可以序列化，那么他们在下次读取时，就是你上次的赋值

public int a; // 可以在程序中赋值，也可以在面板中赋值

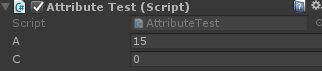
[HideInInspector]

public int b; // 可以在程序中赋值，但在面板上看不到

[SerializeField]

private int c; // 面板中可以看到，而且可以序列化

private int d; // 面板中看不到



**[AddComponentMenu("WolfsGirl/AttributeTest")]**

public class AttributeTest : MonoBehaviour {

会让Component菜单下出现你自定义的类，位置是WolfsGirl/AttributeTest，这样就可以通过这个菜单给其他GameObject添加该组件

**[ExecuteInEditMode]**这个用来修饰类，会自动执行一次

<https://blog.csdn.net/u013616456/article/details/50634796>

[ExecuteInEditMode]

public class AttributeTest : MonoBehaviour

{

public int a; // 可以在程序中赋值，也可以在面板中赋值

[HideInInspector]

public int b; // 可以在程序中赋值，但在面板上看不到

[SerializeField]

private int c; // 面板中可以看到，而且可以序列化

private int d; // 面板中看不到

void Awake()

{

Debug.Log("attribute test awake");

}

// Use this for initialization

void Start()

{

Debug.Log("attribute test start");

}

// Update is called once per frame

void Update()

{

Debug.Log("attribute test update");

}

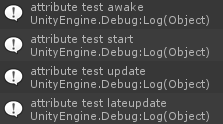
void LateUpdate()

{

Debug.Log("attribute test lateupdate");

}

}

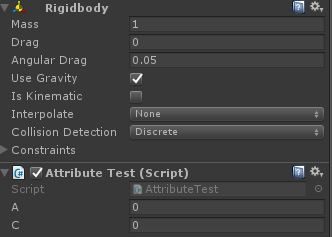


1、当该MonoBehavior在编辑器中被赋于给GameObject的时候，Awake, Start 将被执行。  
2、当Play按钮被按下游戏开始以后，Awake, Start 将被执行。  
3、当Play按钮停止后，Awake, Start将再次被执行。  
4、当在编辑器中打开包含有该MonoBehavior的场景的时候，Awake, Start将被执行

**[RequireComponent(typeof(ClassName))]**就是在你吧把这个脚本拖到GameObject时，自动再给你加上”ClassName”这个类

**[RequireComponent (typeof(Rigidbody))]**

public class AttributeTest : MonoBehaviour



**[ContextMenu(“XXX”)]**在Inspector面板中，右击包含这条标记的类，在菜单中会出现名为XXX的选项，点击该选项，会执行被标记的功能

[ContextMenu ("Test")]

void Test()

{

Debug.Log("This is Test method。");

}



点击右上角的图标，下拉框中会出现Test方法

**[MenuItem(“XXX”)]**添加主菜单，只能定义方法

[MenuItem("Custom/Build AssetBundle")]

public static void Menu()

{

Debug.Log("This is menu method.");

}

点击主菜单的相应按钮会执行该方法，注意：在这里必须要**import UnityEditor**