**约定**

* ShaderLab中, 数学上是Column Vector; 访问接口上是Row-Major
* Object space, World space, Clip space是左手坐标系, View space是右手坐标系.

**Unity判断逐像素/逐顶点处理的光源的规则:**

* 场景中最亮的平行光按照逐像素处理.
* 渲染模式设置为NotImportant的光源按照逐顶点或者SH处理.
* 渲染模式为Important的光源, 按照逐像素处理.
* 如果上述规则得到的逐像素光源小于QualitySetting中的逐像素光源数目(PixelLightCount), 会有更多光源按照逐像素处理.

**渲染设置：**

* Base Pass:   
  Tags { “LightMode = “ForwardBase”}  
  #pragma multi\_compile\_fwdbase  
  该Pass计算环境光, 最重要的平行光, 逐顶点/SH光源和Lightmaps.
* Additional Pass：  
  Tags { “LightMode” = “ForwardAdd” }  
  #pragma multi\_compile\_fwdadd  
  该Pass计算额外的逐像素光源, 每个Pass对应一个光源.

默认情况下不支持阴影, 可以使用下列命令代替上述命令:

#pragma multi\_compile\_fwdadd\_fullshadows

开启.

**关于渲染设置的说明:**

* 环境光和自发光在BasePass中计算(因为每个BasePass只运行一次, 可以定义多个BasePass, 如双面渲染).
* AdditionalPass执行Blend One One, 使pass之间的光照叠加.

**内置函数：**

* float4 UnityObjectToClipPos(float3 p\_ObjSpace);
* float3 UnityObjectToViewPos(float3 p\_ObjSpace);
* float3 WorldSpaceViewDir(float4 v\_ObjSpace);
* float3 ObjSpaceViewDir(float4 v\_ObjSpace);
* float3 WorldSpaceLightDir(float4 v\_ObjSpace);
* float3 ObjSpaceLightDir(float4 v\_ObjSpace);
* float4 ComputeScreenPos(float4 clipPos\_ClipSpace);
* float4 ComputeGrabScreenPos(float4 clipPos\_ClipSpace);