

// MinTuts/Procedural Terrain.shader

```
Shader "MinTuts/Procedural Terrain" {  
    SubShader {  
        Pass {  
            CGPROGRAM  
  
            #pragma vertex   vert  
            #pragma fragment frag  
  
            #include "UnityCG.cginc"  
  
            struct v2f {  
                float4 pos   : SV_POSITION;  
                float3 wpos  : POSITION1;  
            };  
  
            v2f vert(float4 vertex : POSITION) {  
                v2f o;  
  
                o.pos   = UnityObjectToClipPos(vertex);  
                o.wpos  = mul(unity_ObjectToWorld, vertex);  
  
                return o;  
            }  
  
            float4 frag(v2f i) : COLOR {  
                float  p = i.wpos.y * 0.015;  
                float3  y = float3(p, p, p);  
  
                return float4(y, 1);  
            }  
  
        }  
    }  
}
```

Here we are defining the **frag** function...
specified by the **#pragma** definition
from earlier

// MinTuts/Procedural Terrain.shader

```
Shader "MinTuts/Procedural Terrain" {
    SubShader {
        Pass {
            CGPROGRAM

            #pragma vertex    vert
            #pragma fragment frag

            #include "UnityCG.cginc"

            struct v2f {
                float4 pos    : SV_POSITION;
                float3 wpos   : POSITION1;
            };

            v2f vert(float4 vertex : POSITION) {
                v2f o;

                o.pos    = UnityObjectToClipPos(vertex);
                o.wpos   = mul(unity_ObjectToWorld, vertex);

                return o;
            }

            float4 frag(v2f i) : COLOR {
                float  p = i.wpos.y * 0.015;
                float3 y = float3(p, p, p);

                return float4(y, 1);
            }

        }
    }
}
```

Here we are defining the **frag** function...
specified by the **#pragma** definition
from earlier

This function has a single argument...