

## 🎯 16. 结构体作为WGSL函数参数

本节课给大家讲解，在WebGPU案例中，把结构体作为WGSL顶点着色器和片元着色器主函数main的参数。

下面直接在2.14节代码基础上给大家演示。

### 结构体作为顶点着色器main函数参数

下面两端代码的功能是相同的。

顶点位置数据、顶点颜色数据对应的变量作为WGSL顶点着色器main函数的参数，输入给顶点着色器使用。

```
struct Out {  
    @builtin(position) position : vec4<f32>,  
    // vColor表示顶点颜色插值后，每个片元对应的颜色数据  
    @location(0) vColor: vec3<f32>  
}  
@vertex  
// main函数输入顶点位置数据和顶点颜色数据  
fn main(@location(0) pos: vec3<f32>, @location(1) color: vec3<f32>) -> Out {  
    var out: Out;  
    out.position = vec4<f32>(pos, 1.0);  
    out.vColor = color;  
    return out;  
}
```

js

声明一个结构体Input，包含顶点位置数据和顶点颜色数据，然后结构体作为main函数参数。

```
struct Out {  
    @builtin(position) position : vec4<f32>,  
    // vColor表示顶点颜色插值后，每个片元对应的颜色数据  
    @location(0) vColor: vec3<f32>  
}  
struct Input{  
    @location(0) pos: vec3<f32>,  
}
```

js

```

    @location(1) color: vec3<f32>
}
@vertex
// 结构体作为main函数参数: 结构体包含顶点位置数据和顶点颜色数据
fn main(input:Input) -> Out {
    var out: Out;
    out.position = vec4<f32>(input.pos,1.0);
    out.vColor = input.color;
    return out;
}

```

## 结构体作为片元着色器main函数参数

插值后的顶点颜色数据对应变量vColor作为片元着色器main函数的参数。

```

@fragment
// 插值后顶点颜色数据, 作为函数参数
fn main( @location(0) vColor: vec3<f32>) -> @location(0) vec4<f32> {
    // 插值后顶点颜色数据作为赋值给每个片元
    return vec4<f32>(vColor, 1.0);
}

```

js

结构体作为片元着色器main函数的参数。

```

struct Input{
    @location(0) vColor: vec3<f32>
}
@fragment
// 插值后顶点颜色数据, 作为函数参数
fn main( input:Input) -> @location(0) vec4<f32> {
    // 插值后顶点颜色数据作为赋值给每个片元
    return vec4<f32>(input.vColor, 1.0);
}

```

js

