

# ShaderlabVSCode

---

[ShaderlabVSCode](#)

[Introduction](#)

[Installation](#)

[Running On Mac](#)

[Running On Windows](#)

[Features](#)

[Syntax Highlighting](#)

[Code Completion and Basic Intellisense](#)

[Hover Information](#)

[Signature Help](#)

[Document Symbols](#)

[Code Snippets](#)

[Auto Format](#)

[Format Document](#)

[Misc Features](#)

[1. Region Mark](#)

[Features in Unity Editor](#)

[Download Visual Studio Code](#)

[Update Data of ShaderlabVSCode Extension](#)

[Report an Issue](#)

[Release Notes](#)

[V1.1.2](#)

[V1.1.1](#)

[V1.1.0](#)

[V1.0.9](#)

[V1.0.8](#)

[V1.0.7](#)

[V1.0.6](#)

[V1.0.5](#)

[V1.0.4](#)

[V1.0.3](#)

[V1.0.2b3](#)

[V1.0.1b2](#)

[v1.0.0b1](#)

[For more information](#)

## Introduction

---

ShaderlabVSCode is a Visual Studio Code extension for Unity Shaderlab programming.

[Documents](#) | [Forum](#) | [Email](#) | [Asset Store](#)

## Installation

---

---

## Running On Mac

1. Import ShaderlabVSCode unity package into Unity Editor.
2. [Download Visual Studio Code](#) for macOS.
3. Double-click on the downloaded archive to expand the contents.
4. Drag `Visual Studio Code.app` to the `Applications` folder, making it available in the Launchpad.
5. Launch VS Code, Open the `Command Palette (⇧⌘P)` and type 'install from vsix' and then press `Enter` key on keyboard.
6. Select the vsix file under `ShaderlabVSCode/VSCodePlugin/` folder of Unity Project
7. Restart Visual Studio Code

## Running On Windows

1. Import ShaderlabVSCode unity package into Unity Editor.
2. Download the [Visual Studio Code installer](#) for Windows.
3. Once it is downloaded, run the installer (VSCodeSetup-version.exe). This will only take a minute.
4. By default, VS Code is installed under C:\Program Files (x86)\Microsoft VS Code for a 64-bit machine.
5. Launch VS Code, Open the `Command Palette (CTRL+SHIFT+P)` and type 'install from vsix' and then press `Enter` key on keyboard.
6. Select the vsix file under `ShaderlabVSCode/VSCodePlugin/` folder of Unity Project
7. Restart Visual Studio Code

Note: .NET Framework 4.5.2 is required for VS Code. If you are using Windows 7, please make sure .NET Framework 4.5.2 is installed.

## Features

---

### Syntax Highlighting

```
NewUnlitShader.shader x
1 Shader "Unlit/NewUnlitShader"
2 {
3     Properties
4     {
5         _MainTex ("Texture", 2D) = "white" {}
6     }
7     SubShader
8     {
9         Tags { "RenderType"="Opaque" }
10        LOD 100
11
12        Pass
13        {
14            CGPROGRAM
15            #pragma vertex vert
16            #pragma fragment frag
17            // make fog work
18            #pragma multi_compile_fog
19
20            #include "UnityCG.cginc"
21
22            struct appdata
23            {
24                float4 vertex : POSITION;
25                float2 uv : TEXCOORD0;
26            };
```

## Code Completion and Basic Intellisense

```
fixed4 frag (v2f i) : SV_Target
{
    // sample the texture
    fixed4 col = tex2D(_MainTex, i.uv);
    // apply fog
    UNITY_APPLY_FOG(i.fogCoord, col);
    i.
    localPos float4 x
    if uv float4 localPos: TEXCOORD1
    { vertex
      discard;
    }
    else
    {
        return col;
    }
}
```

## Hover Information

```
float _Offset;
v2f vert (appda
{
    v2f o;
    o.localPos
    o.vertex = UnityObjectToClipPos(v.vertex);
    o.uv = TRANSFORM_TEX(v.uv, _MainTex);
    UNITY_TRANSFER_FOG(o,o.vertex);
    return o;
}
```

Transforms a point from object space to the camera's clip space in homogeneous coordinates. This is the equivalent of `mul(UNITY_MATRIX_MVP, float4(pos, 1.0))`, and should be used in its place.

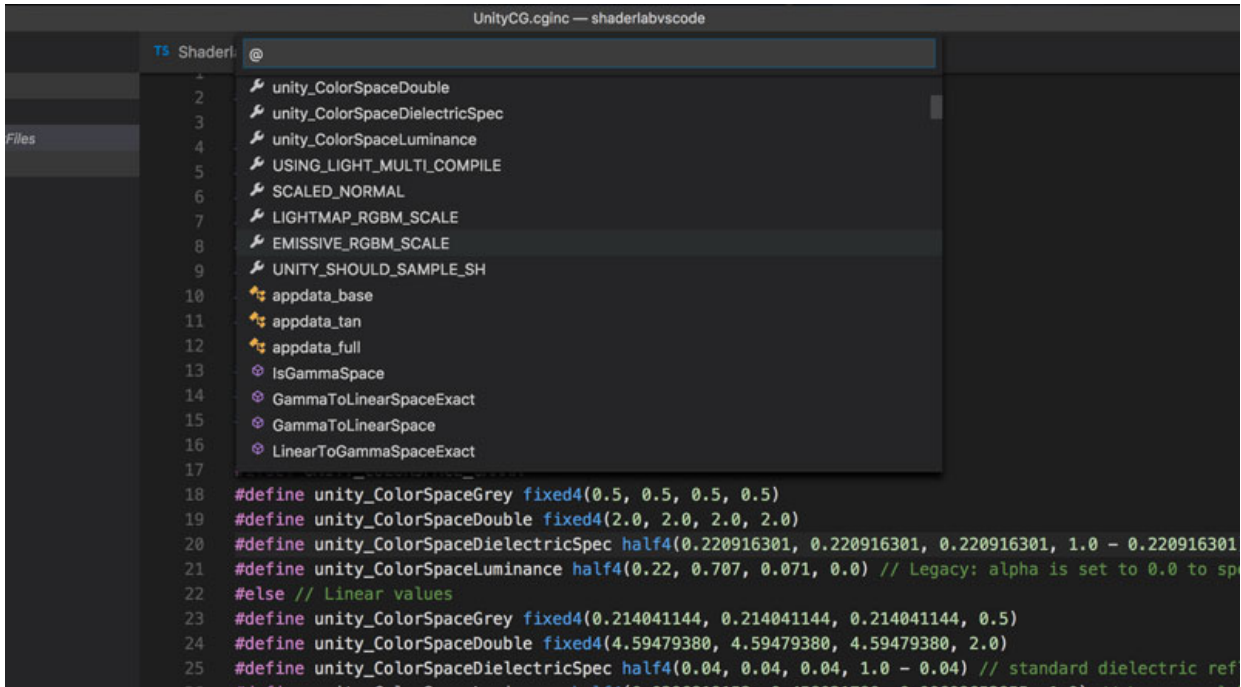
## Signature Help

```
v2f vert (appdata v)
{
    v2f o;
    o.localPos = v.vertex;
    o.vertex = UnityObjectToWorldUnits(v.vertex);
    o.uv = TRANSFORM_TEX(v.uv, _MainTex);
    o.localPos = normalize(o.vertex - o.localPos);
    UNITY_TRANSFER_FOG(o,o.vertex);
    return o;
}
```

`float normalize(float v)`  
normalizes a vector

## Document Symbols

Press `CTRL + SHIFT + o` on Windows or `CMD + SHIFT + o` on macOS.



## Code Snippets

Below are the snippets:

Snippets	Description
blend1_1	Blend One One
blendsa_1-sa	Blend SrcAlpha OneMinusSrcAlpha
blend1_1-sa	Blend One OneMinusSrcAlpha
blend1-dc_1	Blend OneMinusDstColor One
blenddc_0	Blend DstColor Zero
blenddc_sc	Blend DstColor SrcColor
cgp	CGPROGRAM...ENCG

for	for loop
fallback	Fallback
glp	GLSLPROGRAM...ENCGLSL
if	if { ... }
ifelse	if {...} else {...}
incucg	#include "UnityCG.cginc"
inclight	#include "Lighting.cginc"
incautolight	#include "AutoLight.cginc"
props	Properties
prop2d	<span>2D</span> type property
propcube	<span>Cube</span> type property
propc	<span>Color</span> type property
propv	<span>Vector</span> type property
propf	<span>Float</span> type property
proprange	<span>Range</span> type proprety
region	//#region ... //endregion
region2	//region ... //endregion
shader	Shader { ... }
subshader	SubShader { ... }
struct	structure
tags	Tags { ... }
tagstt	Tags with both of RenderType and Queue is Transparent

## Auto Format

### Format Document

Two ways to format document:

1. Right click the editor are and select **Format Document** menu in context menu
2. Open **Command Palette** and type "Format Document", and then press ENTER key on keyborad.

# Misc Features

## 1. Region Mark

There are two ways:

- `///#region` and `///#endregion`
- `///region` and `///#endregion`

## Features in Unity Editor

### Download Visual Studio Code

Jump to url which can download latest version of Visual Studio Code

Selection: **Tools** -> **ShaderlabVSCode** -> **Download Visual Studio Code**

### Update Data of ShaderlabVSCode Extension

Update data of completion, hover information or intelisense from web

Selection: **Tools** -> **ShaderlabVSCode** -> **Update Data of VSCode Extension**

### Report an Issue

Two ways to report an issue:

1. Send Email to [amlovey@qq.com](mailto:amlovey@qq.com)
2. Open a issue on <https://github.com/amloveyweb/amloveyweb.github.io/issues>

## Release Notes

---

### V1.1.2

- Add more completions from UnityCG.cginc.
- Fix document symbols show incorrectly in some scenarios.

### V1.1.1

- Intellisense
  - Add macros code completion support
  - Add more completion items from UnityCG.cginc, there are:
    - UnityWorldSpaceViewDir
    - UnityWorldToClipPos
    - UnityViewToClipPos
    - UnityWorldToViewPos
    - UnityObjectToWorldDir
    - UnityWorldToObjectDir

- UnityObjectToWorldNormal
  - UnityWorldSpaceLightDir
- Fix methods intellisense was broken by ':' in parameters
- Format Document
  - Improve format for macros
- Syntax Highlighting
  - Add highlight for `#ifdef` and `#ifndef`
- Add document symbols support, press `CTRL + SHIFT + o` on Windows or `CMD + SHIFT + o` on macOS to open it.

## V1.1.0

- Intellisense
  - Fix intellisense was broken by "," in structs and fields in some scenario
- Format Document
  - Fix format for `[XX]PROGRAM..END[XX]` structure
  - Make `#define` to match levels
- Experiment:
  - Add `.hls` and `.cg` file support

## V1.0.9

- Intellisense
  - Fix wrong result when there are duplicate name of variables
- Syntax Highlighting
  - Add highlighting for custom functions

## V1.0.8

- Format Document
  - Improve format for Operators

## V1.0.7

- Intellisense
  - Fix Intellisense broken by '+', '-', '\*', '/' in some scenarios
- Format Document
  - Improve format for preprocessor directives
- Syntax Highlighting
  - Improve color of preprocessor directives

## V1.0.6

- Format Document:
  - fix colon formation is incorrect in #pragma line
- Intellisense
  - Fix wrong code completion result in #pragma line which is triggered by colon
  - Update description for `clip` and `cos` cg method in code completion item

## V1.0.5

- Add region mark support(Required VSCode version 1.17.0 +). ShaderlabVSCode now supports two type markers:
  - `//#region` and `//#endregion`, snippet is `region`
  - `//region` and `//endregion`, snippet is `region2`

## V1.0.4

- Intellisense
  - Add Unity defined Values support, like `_Time`
  - Fix duplicate members when include same cginc files multiple times

## V1.0.3

- Intellisense
  - Fix bug variable broken by semicolon
- Editor
  - Improve compability

## V1.0.2b3

- Auto Format:
  - Add format document feature
- Intellisense:
  - supports builtin types, like half, fixed and float
  - supports completion of fields of types
  - supports completion of method return type
- Bug Fixes:
  - Fix bug structure fields are broken by comments

## V1.0.1b2

- Add code snippets support
- Update hover information for some keywords
- Fix wrong fields data get from struct in some scenarios



- Fix bug that Variable and Properties Info broken by whitespace

## **v1.0.0b1**

- First beta release

---

## **For more information**

---

Visit site <http://www.amlovey.com>