

EmmyLua注解功能

EmmyLua 编辑器插件利用注解 `annotation` 功能来增加代码提示以及其它编辑器功能

1. @class类声明注解

EmmyLua 利用 `@class` 注解来模拟面向对象中的类，可以继承，可以定义字段/属性

- 完整格式：

```
--@class {my_type}[: parent_type] @comment string
```

- 应用目标：

```
local 变量  
global 变量
```

- 示例：

```
---@class Car : Transport @定义一个Car类继承父类  
Transport  
local cls = class()  
  
function cls:test()  
end
```

- 示例说明：

将 `cls` 变量标记为 `Car` 类，在其它地方可以使用 `@type` 注解来标记目标变量类型，以增强代码提示以及其它功能

2. @type类型标记注解

利用 `@type` 注解来标记目标变量的类型，以增强代码提示以及其它功能

- 完整格式：

```
---@type {my_type}[/other_type] @comment string
```

- 应用目标：

```
local 变量
```

`global` 变量

- 示例:

```
---@type Car @汽车的实例
local car1 = {}

---@type Car|Ship @交通工具，车或者船。因lua脚本的灵活性，一个变量可能对应多个类型，用|符号表示可能的类型
local transport = {}

---@type Car @标记全局的变量类型
global_car = {}
```

3. @param参数类型标记注解

利用 `@param` 注解来标记函数定义参数的类型，以增强代码提示以及其它功能

- 完整格式:

```
---@param {param_name} {my_type}[|other_type]
@comment string
```

- 应用目标:

函数参数
for 循环参数

- 示例:

```
---@param car Car
local function setCar(car)
end

---@param car Car
for k, car in ipairs(list) do
end
```

4. @return 函数返回值注解

利用 `@return` 注解来标记函数的返回值类型

- 完整格式:

```
---@return {my_type}[|other_type] @comment string
```

- 应用目标:

函数

- 示例:

```
---@return Car|Ship
local function create()
    ...
end

---此时car_or_ship类型可以不用@type标记, EmmyLua已通过create函数的推断出了类型
local car_or_ship = create()
```

```
---@return Car
function factory:create()
    ...
end
```

5. @field 属性注解

利用 `@field` 注解来标记某个类的额外的属性（即使这个属性没有出现在代码里）

- 完整格式:

```
---@field public|protected {my_field} {field_type}[|other_type]
@comment string
```

- 应用目标:

有类 `@class` 注解的地方

- 示例:

```
---@class Car
---@field public name string @标记Car有一个name属性, 代码提示中会出现相应提示
local cls = class()
```

6. 完整示例

```
---@class Transport @父类
---@public name string
local transport = {}

function transport:move()end

---@class Car : Transport @Car继承自Transport
local car = {}
function car:move()end

---@class Ship : Transport @Ship继承自Transport
local ship = {}

---@param type number @参数type说明
---@return Car|Ship @返回类型可能是Car也有可能是Ship
local function create(type)
    -- 略
end

local obj = create(1)
---此时obj可代码提示

---@type Car
local obj2
---此时obj2可代码提示

local list = { obj, obj2 }
---@param v Transport
for _, v in ipairs(list) do
    ---此时v可代码提示
end
```

7. 最后说明

EmmyLua注解功能只是单纯的辅助编辑器代码提示以及其它功能，和Lua代码的实际运行逻辑没有任何关系，因为它们就是普通的Lua注释