转载自https://www.cnblogs.com/aspiration2016/p/8433122.html

Doxygen是一种开源跨平台的，以类似JavaDoc风格描述的文档系统，完全支持C、C++、Java、Objective-C和IDL语言，部分支持PHP、C#。鉴于Doxygen良好的注释风格，故基于Doxygen以形成自己的注释规范。

1 标注总述

```
//---------------------------------------------------------------------
// Platform Defines
//---------------------------------------------------------------------
enum
{
    OST_PLATFORM_WIN32          = 1,
    OST_PLATFORM_LINUX_X86      = 2,
    OST_PLATFORM_LINUX_ARM      = 3,
    OST_PLATFORM_ANDROID        = 4,
    OST_PLATFORM_MACOSX         = 5,
};

//---------------------------------------------------------------------
// API Export/Import Macros
//---------------------------------------------------------------------
/** Indicates an exported and imported shared library function. */
#define OST_API_EXPORT         __declspec(dllexport)
#define OST_API_IMPORT         __declspec(dllimport)

//---------------------------------------------------------------------
// Digital Image Macros
//---------------------------------------------------------------------
#define OST_PI                          3.141592653589793f
#define OST_RGB2GRAY(r, g, b)         ( ((b) * 117 + (g) * 601 + (r) * 306) >> 10 )

//---------------------------------------------------------------------
// date and time at compile time
//---------------------------------------------------------------------
#define OST_TIMESTAMP                 __DATE__ " " __TIME__
```

2 文件头的标注

```
/*****************************************************************************
*  OpenST Basic tool library                                                 *
*  Copyright (C) 2014 Henry.Wen  renhuabest@163.com.                         *
*                                                                            *
*  This file is part of OST.                                                 *
*                                                                            *
*  This program is free software; you can redistribute it and/or modify      *
*  it under the terms of the GNU General Public License version 3 as         *
*  published by the Free Software Foundation.                                *
*                                                                            *
*  You should have received a copy of the GNU General Public License         *
*  along with OST. If not, see <http://www.gnu.org/licenses/>.               *
*                                                                            *
*  Unless required by applicable law or agreed to in writing, software       *
*  distributed under the License is distributed on an "AS IS" BASIS,         *
*  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  *
*  See the License for the specific language governing permissions and       *
*  limitations under the License.                                            *
*                                                                            *
*  @file     Example.h                                                       *
*  @brief    对文件的简述                                                  *
*  Details.                                                                  *
*                                                                            *
*  @author   Henry.Wen                                                       *
*  @email    renhuabest@163.com                                              *
*  @version  1.0.0.1(版本号)                                                 *
*  @date     renhuabest@163.com                                              *
*  @license  GNU General Public License (GPL)                                *
*                                                                            *
*----------------------------------------------------------------------------*
*  Remark         : Description                                              *
*----------------------------------------------------------------------------*
*  Change History :                                                          *
*  <Date>     | <Version> | <Author>       | <Description>                   *
*----------------------------------------------------------------------------*
*  2014/01/24 | 1.0.0.1   | Henry.Wen      | Create file                     *
*----------------------------------------------------------------------------*
*                                                                            *
*****************************************************************************/
```

3 命名空间

```
/**
 * @brief 命名空间的简单概述 \n(换行)
 * 命名空间的详细概述
 */
namespace OST
 {
 }
```

4 类、结构、枚举标注

```
/**
 * @brief 类的简单概述 \n(换行)
 * 类的详细概述
 */
class Example
 {
 };
```

枚举类型定义、结构体类型定义注释风格类似

```
/**
 * @brief 简要说明文字
 */
typedef struct 结构体名字
 {
      成员1, /*!< 简要说明文字 */ or ///<说明,  /**<说明 */
      成员2, /*!< 简要说明文字 */ or ///<说明,  /**<说明 */
      成员3, /*!< 简要说明文字 */ or ///<说明,  /**<说明 */
 }结构体别名;
```

5 函数注释原则

```
/**
 * @brief 函数简要说明-测试函数
 * @param index    参数1
 * @param t        参数2 @see CTest
 *
 * @return 返回说明
 *     -<em>false</em> fail
 *     -<em>true</em> succeed
 */
bool Test(int index, const CTest& t);
```

note：指定函数注意项事或重要的注解指令操作符 note格式如下： @note 简要说明

retval：指定函数返回值说明指令操作符。(注:更前面的return有点不同.这里是返回值说明) retval格式如下： @retval 返回值 简要说明

pre：指定函数前置条件指令操作符 pre格式如下： @pre 简要说明

par：指定扩展性说明指令操作符讲。（它一般跟code、endcode一起使用 ） par格式如下： @par 扩展名字

code、endcode：指定 code、endcode格式如下： @code 简要说明(内容) @endcode

see：指定参考信息see格式如下： @see 简要参考内容

deprecated：指定函数过时指令操作符。 deprecated格式如下： @deprecated 简要说明

调试Bug说明

解决的bug说明，@bug 警告说明

(warning) 定义一些关于这个函数必须知道的事情，@warning 备注说明

(remarks) 定义一些关于这个函数的备注信息，@remarks 将要完成的工作

(todo) 说明哪些事情将在不久以后完成，@todo 使用例子说明

(example) 例子说明，@example example.cpp

```
/**
* @brief 打开文件 \n
* 文件打开成功后，必须使用::CloseFile函数关闭
```

```
* @param[in] fileName     文件名
* @param[in] fileMode     文件模式，可以由以下几个模块组合而成：
*      -r读取
*      -w 可写
*      -a 添加
*      -t 文本模式(不能与b联用)
*      -b 二进制模式(不能与t联用)
* @return 返回文件编号
*  --1表示打开文件失败(生成时:.-1)
* @note文件打开成功后，必须使用::CloseFile函数关闭
* @par 示例:
* @code
*         //用文本只读方式打开文件
*         int ret = OpenFile("test.txt", "a");
* @endcode
* @see 函数::ReadFile::CloseFile (“::”是指定有连接功能,可以看文档里的CloseFile变成绿,点击它可以跳转到CloseFile.)
* @deprecated由于特殊的原因，这个函数可能会在将来的版本中取消
*/
int OpenFile(const char* fileName, const char* fileMode);

/**
* @brief 关闭文件
* @param [in] file    文件
*
* @retval 0      成功
* @retval -1     失败
* @pre file 必须使用OpenFile的返回值
*/
int CloseFile(int file);
```

-：生成一个黑心圆. -#：指定按顺序标记。 ::：指定连接函数功能。（注：空格和“:”有连接功能,但建议还是使用”::”。只对函数有用。） 它们格式如下:(-和::例子前面有了,就介绍-#例子。) - 简要说明 -# 简要说明 ::函数名 例：

```
/**
* @param [in] person 只能输入以下参数：
* -# a:代表张三          // 生成 1. a:代表张三
* -# b:代表李四          // 生成 2. b:代表李四
* -# c:代表王二          // 生成 3. c:代表王二
*/
void GetPerson(int p);
```

6 变量注释

```
/// 简述
/** 详细描述. */
```

或者

```
//! 简述
//! 详细描述
//! 从这里开始
int m_variable_1; ///< 成员变量m_variable_1说明
int m_variable_2; ///< 成员变量m_variable_1说明

/**
* @brief 成员变量m_c简要说明
*
* 成员变量m_variable_3的详细说明，这里可以对变量进行
* 详细的说明和描述，具体方法和函数的标注是一样的
*/
bool m_variable_3;
```

如果变量需要详细说明的可已按照m_varibale_3的写法写，注意，m_variable_2和m_variable_3之间一定需要空行，否则会导致m_variable_2的简述消失 7. 模块标注 模块定义格式:

```
/**
* @defgroup 模块名   页的标题名 (模块名只能英文,这个可以随便取.在一个源文件里不能相同)
* @{ (跟c语言{一样起作用域功能)
*/
… 定义的内容 …
/** @} */
```

例：

```
/**
* @defgroup HenryWen Example.cpp
* @{
*/
  … 定义的内容 …
```

```
/** @} */
```

\8. 分组标注 分组定义格式：

```
/**
* @name 分组说明文字
* @{
*/
… 定义的内容 …
/** @} */
```

例：

```
/**
* @name PI常量
* @{
*/
#define PI 3.1415926737
/** @} */

/**
* @name 数组固定长度常量
* @{
*/
const int g_ARRAY_MAX = 1024;
/** @} */
```