文档目的：以下为实现合作开发的简单代码命名规范

负责人：邓积懋

时间：2019.6.18

**文档正文**

要解决的问题：

（1）函数的命名

例如例子函数的命名，其功能是输出欢迎语句，故函数名为welcome，若遇到一个动词都无法描述的函数，则采用小驼峰式命名，例如打印结果printResult，获取命令长度getCommandLength等等。

读者只需要记得，函数命名其最重要目的是表达函数的功能（故一般是动词和名词的组合），其次才是区分其他函数。故除非命名的简称被普遍认知，否则最好用全称，例如length的缩写len，control的缩写ctrl。

（2）注释的分布

主要为三种注释。

第一种注释，函数外部入口前，注明负责人，函数的简短描述；

第二种注释，函数内部开始前，对函数的功能，限制等做详细的或者补充的说明；

第三种注释，存在于代码行间，用于对代码的补充说明，由个人自己决定。

为了协调性，三种代码的注释形式尽量采用/\*\*/，少用或不用//。

代码典例

/\*\*\*\*\* Info \*\*\*\*\*/

/\* Auther：DJM \*/

/\* Function：Shell入口，打印欢迎信息 \*/

/\*\*\*\*\* Info \*\*\*\*\*/

void welcome(){

/\* Note：如下是欢迎信息 \*/

/\* 为了是程序更友好，加入了颜色 \*/

/\* 颜色是紫色，背景色与shell相同 \*/

printf("\e[35mwelcome to myshell\e[0m\n"); /\*第一行打印文字\*/

printf("\e[35mit's a shell program made by WuYusong\e[0m\n");

printf("\e[35mhope you have a good time with it :-)\e[0m\n");

}