

## 变量说明：

在Makefile中定义的变量，就像是C/C++语言中的宏一样，他代表了一个文本字符串，在Makefile中执行的时候其会自动原模原样地展开在所使用的地方，变量的命名字可以包含字符、数字，下划线（可以是数字开头），但不应该含有“:”、“#”、“=”或是空字符（空格、回车等）。变量是大小写敏感的，“foo”、“Foo”和“FOO”三个不同的变量名。传统的Makefile的变量名是全大写的命名方式，但我推荐使用大小写搭配的变量名，如：

MakeFlags。这样可以避免和系统的变量冲突，而发生意外的事情。

有一些变量是很奇怪字符串，如“\$<”、“\$@”等，这些是自动化变量

## 一、变量的基础

变量在声明时需要给予初值，而在使用时，需要给在变量名前加上“\$”符号，但最好用小括号“()”或是大括号“{}”把变量给包括起来。如果你要使用真实的“\$”字符，那么你需要用“\$\$”来表示。

变量可以使用在许多地方，如规则中的“目标”、“依赖”、“命令”以及新的变量中。先看一个例子：

```
objects = program.o foo.o utils.o
program : $(objects)
        cc -o program $(objects)
$(objects) : defs.h
```

变量会在使用它的地方精确地展开，就像C/C++中的宏一样，例如：

```
foo = c
prog.o : prog.$(foo)
        $(foo)$(foo) -$(foo) prog.$(foo)
```

展开后得到：

```
prog.o : prog.c
        cc -c prog.c
```

当然，千万不要在你的Makefile中这样干，这里只是举个例子来表明Makefile中的变量在使用处展开的真实样子。可见其就是一个“替代”的原理。

另外，给变量加上括号完全是为了更加安全地使用这个变量，在上面的例子中，如果你不想给变量加上括号，那也可以，但我还是强烈建议你给变量加上括号。