前言：

由于在shell脚本中碰到了这个命令，所以索性将所有的关于冒号的语法都看一遍！

**: ${VAR:=DEFAULT}**当变量VAR没有声明或者为NULL时，将VAR设置为默认值DEFAULT。如果不在前面加上:命令，那么就会把${VAR:=DEFAULT}本身当做一个命令来执行，报错是肯定的。

用途说明

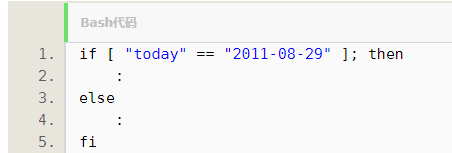
我们知道，在Linux系统中，冒号(:)常用来做路径的分隔符（PATH），数据字段的分隔符（/etc/passwd）等。其实，冒号(:)在Bash中也是一个内建命令，它啥也不做，是个空命令、只起到占一个位置的作用，但有时候确实需要它。当然，它也有它的用途的，否则没必要存在。在·Linux的帮助页中说它除了参数扩展和重定向之外不产生任何作用。

经典用法

**用法1：占位符作用，此时仅输入一个冒号：**

·啥也不做，只起到占位符的作用。比如在编写脚本的过程中，某些语法结构需要多个部分组成，但开始阶段并没有想好或完成相应的代码，这时就可以用:来做占位符，否则执行时就会报错。

示例：



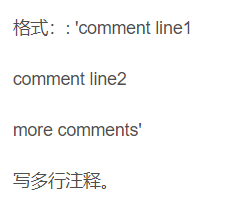
**用法2：单行注释与多行注释**

单行注释

: 注释语句

类似于：# 注释语句

多行注释



**用法3:清空文本file的内容 :>file**

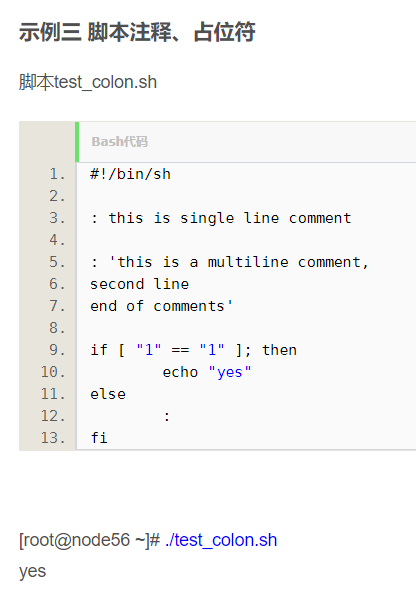
**用法4： : ${VAR:=DEFAULT}，起到赋值给VAR的作用**

当变量VAR没有声明或者为NULL时，将VAR设置为默认值DEFAULT。如果不在前面加上:命令，那么就会把${VAR:=DEFAULT}本身当做一个命令来执行，报错是肯定的。

使用示例：







**用法5： ：- + = ？数字 系列，常出现在${ }中。**

**缺省值（:-）**

如果变量后面跟着冒号和减号，则变量后面跟着是这个变量的缺省值。

$ COMPANY=

$ printf “%s/n” “${COMPANY:-Unknown Company}”

Unknown Company

变量的实际值可以保持不变。

冒号也可以省略掉不用：

$ COMPANY=

$ printf “%s/n” “${COMPANY-Nightlight Inc.}”

$

**指定缺省值（:=）**

如果变量后面跟着冒号和等号，则给空变量指定一个缺省值。

$ printf “%s/n” “${COMPANY:=Nightlight Inc.}”

Nightlight Inc.

$ printf “%s/n” “$COMPANY”

Nightlight Inc.

变量的实际值已经改变了。

去除冒号，则不会指定缺省值。

**变量是否存在检查（:?）**

如果变量后面跟着冒号和问号，则根据变量是否存在，显示不同的信息。信息不是必选的。

printf “Company is %s/n” /

“${COMPANY:?Error: Company has notbeen defined—aborting}”

如果没有冒号则不会进行检查。

**覆盖缺省值(:+)**

如果变量后面跟着冒号和加号，则加好后面的字符串替换默认字符串。

$ COMPANY=”Nightlight Inc.”

$ printf “%s/n” “${COMPANY:+Company has been overridden}”

Company has been overridden

如果没有冒号，变量也被字符串所替换，变量本身的值不改变。

**替换部分字符串（:n）**

如果变量后面跟着一个冒号和数字，则返回该数字开始的剩余子字符串，如果后面还跟着一个冒号和数字。则第一个数字表示开始的字符，后面数字表示字符的长度。

$ printf “%s/n” “${COMPANY:5}”

light Inc.

$ printf “%s/n” “${COMPANY:5:5}”

light

**标出使用范围，即使用范围符号，格式为：[[:punct:]]**

[[:class:]] 通过class指定字符类别，class可以是POSIX标准中的下列关键字：alnum、alpha、ascii、blank、cntrl、digit、graph、lower、print、punct、space、upper、word、xdigit，其中word表示大小写字母、数字和下划线。

示例1：

$ COMPANY=”Nightlight Inc.”

$ printf “%s” “${COMPANY//[[:punct:]]}”

结果：Nightlight Inc

示例2：



**用法6：三元运算符**

