Shell 脚本语言

1：#！定义由什么shell解释 否则默认shell 一般 #！/bin/bash

2：# 表示注释当前这一行

3：定义变量：变量名=变量值 变量名不能包含数字

4：引用变量：$变量名 相当于将变量名换成变量值

5：unset 清除变量值

6：read 从键盘获取值 然后赋值给变量名 如read 变量名 –p 则是可以在获取值时添加提示 类比scanf时前输出了printf 可以读取多个值 和多个变量名对应

7：readonly 获取值后不可更改 就是只有只读权限，之后无法修改

8：env 显示环境变量

9：source 脚本文件

source命令用法:

source FileName

作用:在当前bash环境下读取并执行FileName中的命令。

注:该命令通常用命令“.”来替代。

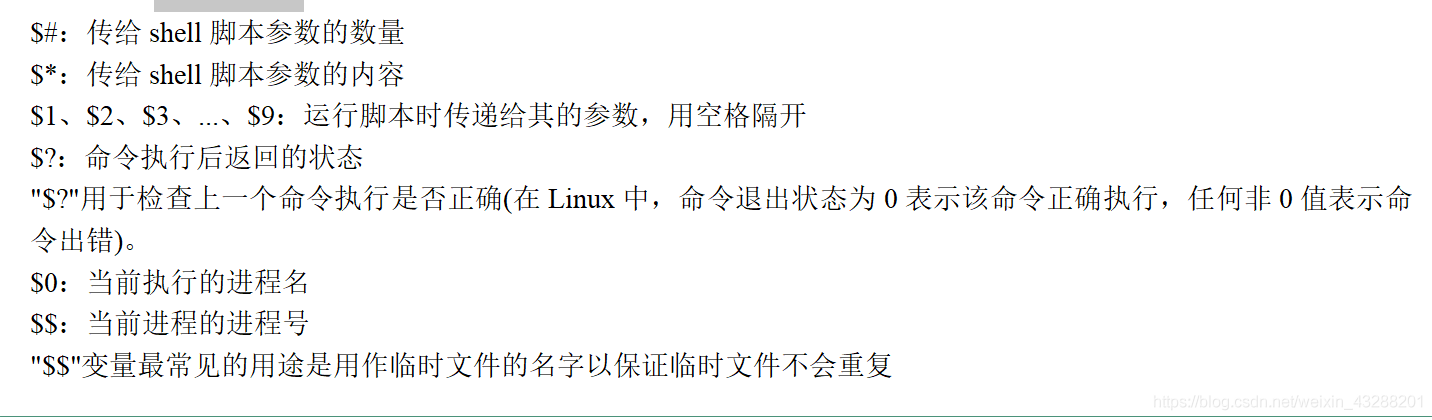
如:source .bash\_rc 与 . .bash\_rc 是等效的。

注意:source命令与shell scripts的区别是，

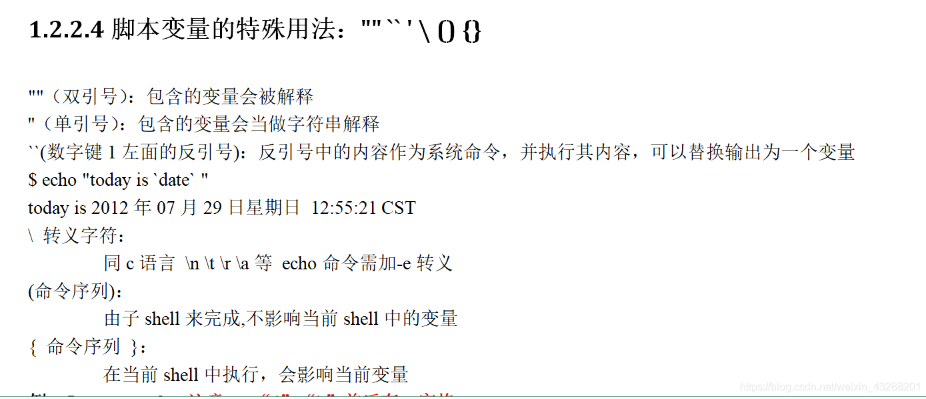
source在当前bash环境下执行命令，而scripts是启动一个子shell来执行命令。这样如果把设置环境变量(或alias等等)的命令写进scripts中，就只会影响子shell,无法改变当前的BASH,所以通过文件(命令列)设置环境变量时，要用source 命令。

10：双引号 单引号的区别 双引号：可以解析变量的值 单引号：不能解析变量的值，此情况下就是将$看为字符串

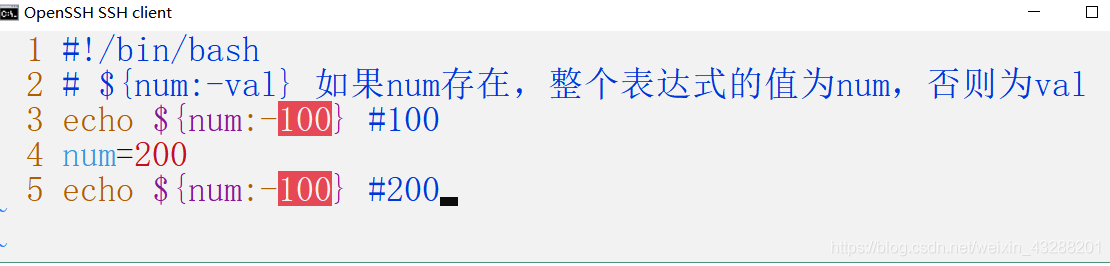
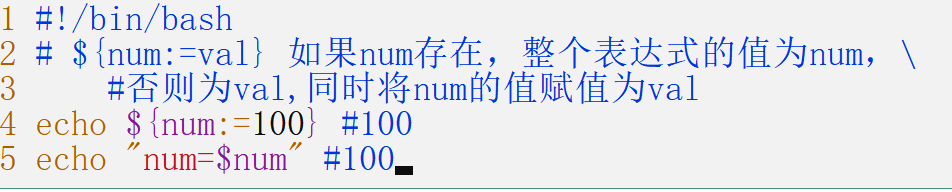
11：临时添加PATH环境变量 export PATH=$PATH:/需要添加的路径 注：这种只是临时的永久需要修改配置文件，自行查找方法，不难

12：

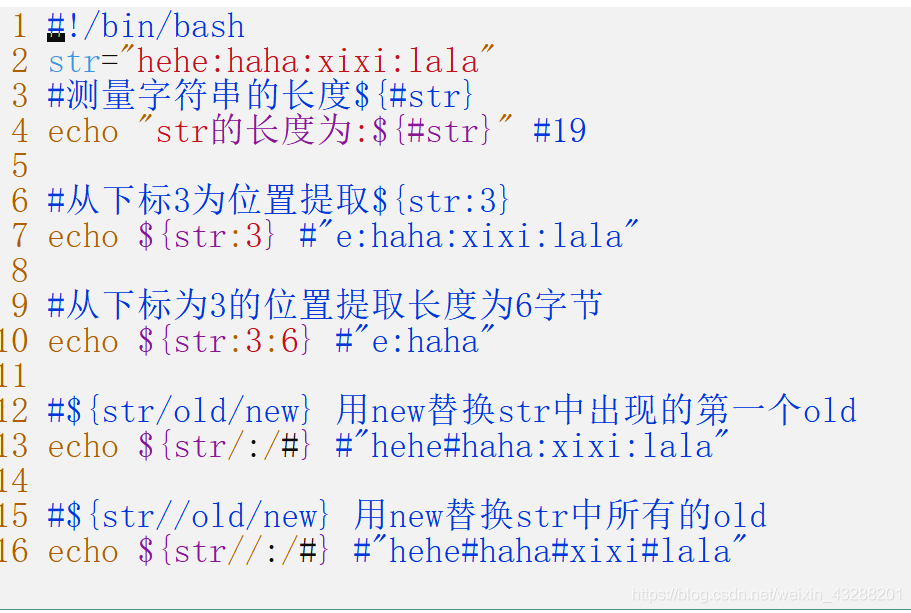
13：



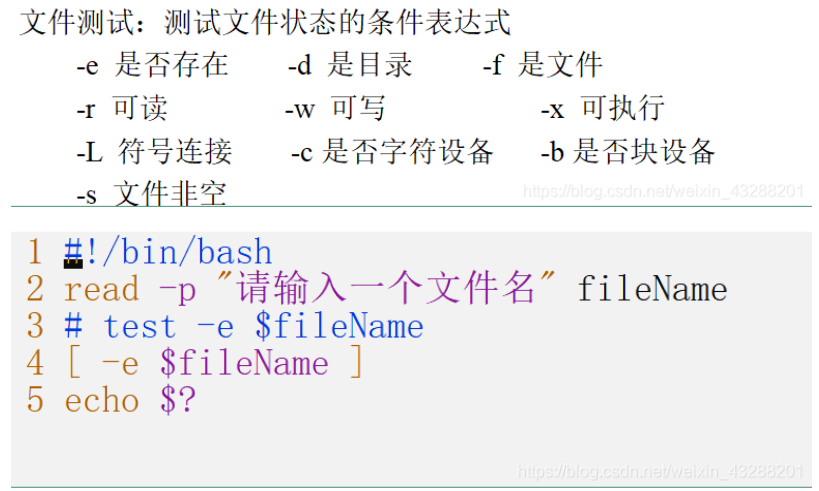
14：判断变量是否存在：



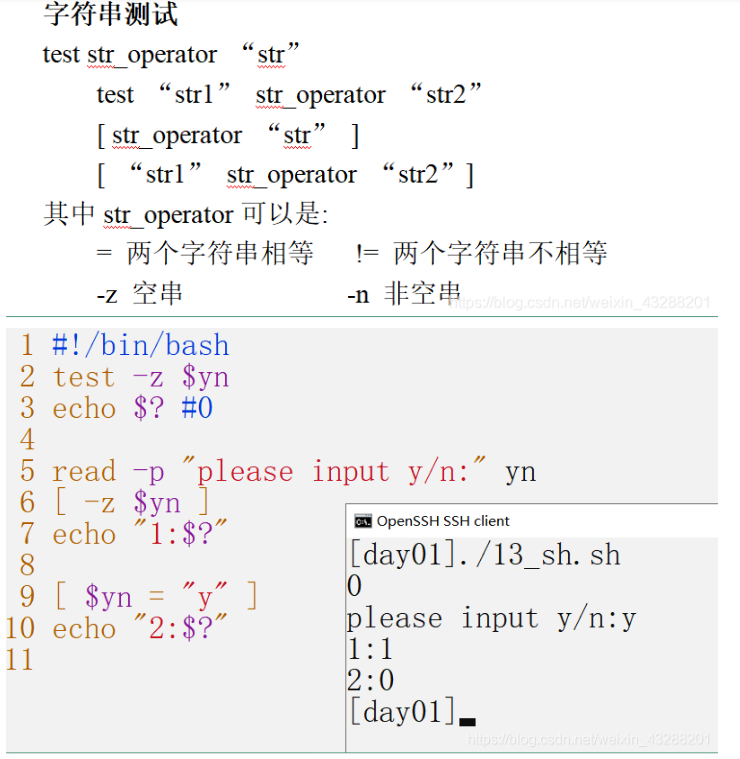
15：对于字符串的操作：



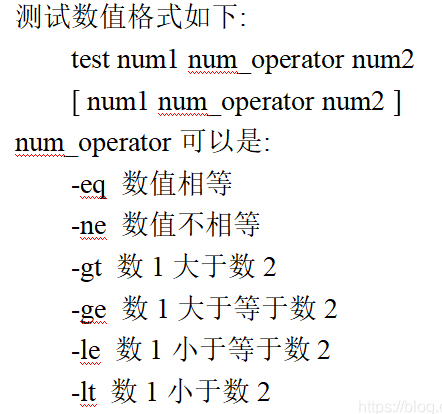
16：文件测试



17：



18：数值测试 和字符串测试差不多 符号有差异

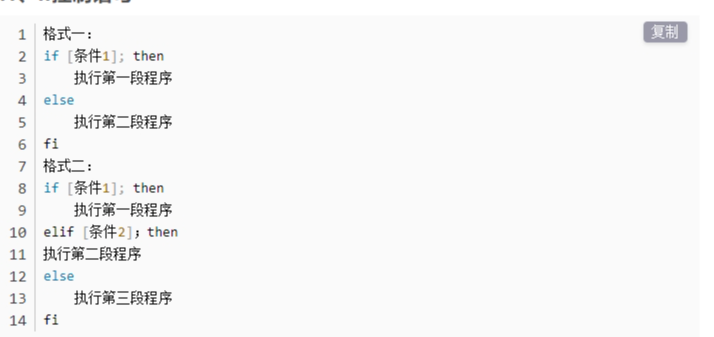


19：与 && 意思有差异 这是前一个命令执行成功才执行后一个命令

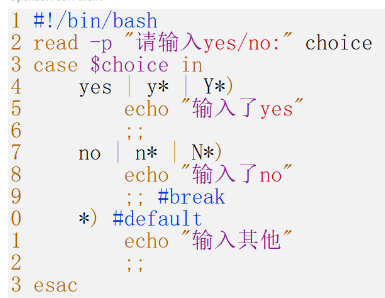
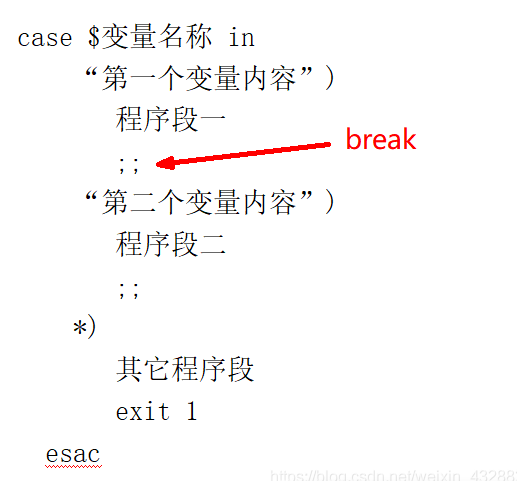
20：或 || 前一个命令执行失败执行后一个命令

21：-a 同时满足返回真 –o 一个满足返回真 ！不满足返回真

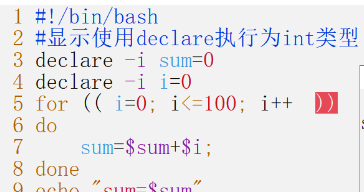
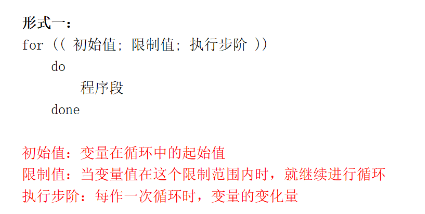
22：判断语句：

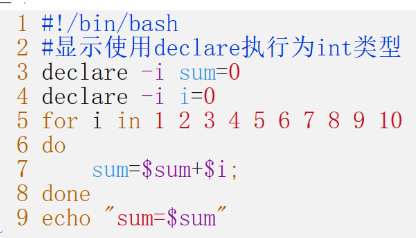
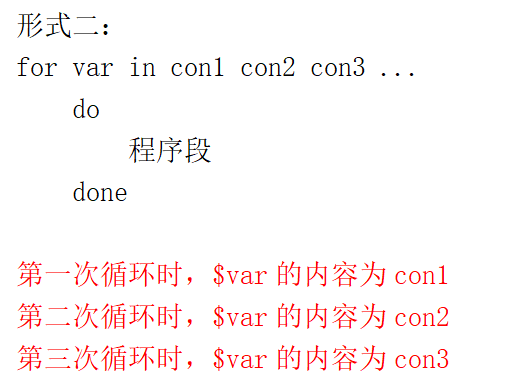


23：case用法

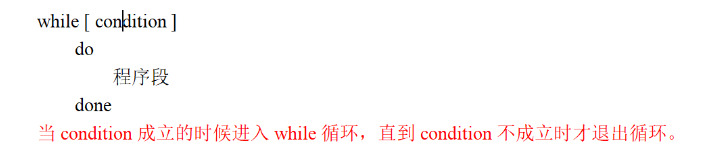


24：for的用法





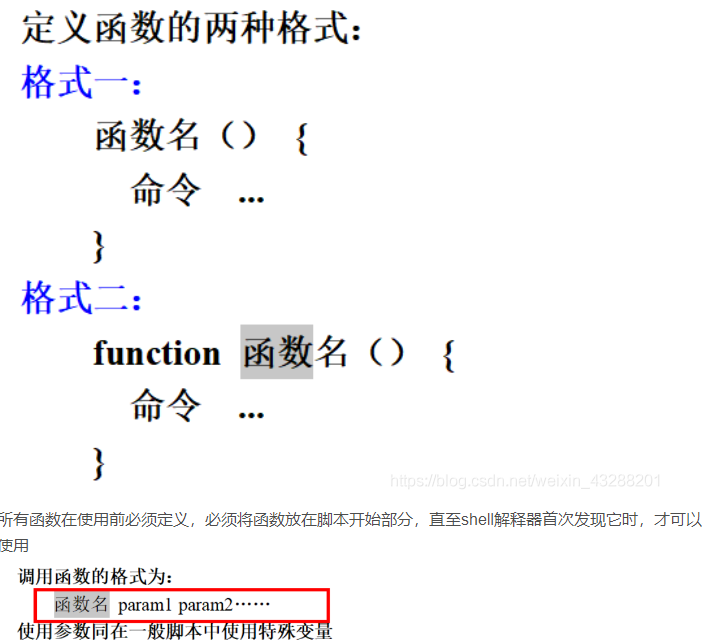
25：while 循环



26：until 与while相反 不成立时进入

27：break continue 和 C语言中定义一样

28：函数



29：return 返回值 和C一样

30：source 文件名 可以导入函数 也可直接调用文件

31：cut 参数 文件名 按一定格式切割文件

本文借鉴大佬文章，可查看原文：[shell脚本语言(超全超详细)\_看见代码就想敲的博客-CSDN博客](https://blog.csdn.net/weixin_43288201/article/details/105643692)

谷歌搜索技巧 ：

1：用双引号包括关键词 可以保证完全匹配，增加精确度

2：用~排除内容 将会使搜索结果中的一些内容不再显示 尤其是排除来自什么来源时，如排除维基百科内容 ~wikipedia

3：使用\*通配，适合忘记某个单词时使用

4：用…来表示范围 如 2021…2022 表2021至2022

5：search: 域名 内容 搜索指定域名中的内容

6：使用filetype: 来查找指定类型的文件 如 filetype:pdf 查找pdf文件

7：查找多个关键词 AND

8：查找其中一个 OR

9：查找关键词的定义 Define : 如Define: Javascript

10：搜索相关的关键词 +

11：查找相似的网站 related: 如 related: [http://w3schools.com](https://link.zhihu.com/?target=http%3A//w3schools.com)

12：site:域名 内容 和search类似 查找指定站点的内容

13：搜索指定时间范围：after: before: