**bash变量**

bash中变量有两种，分别是环境变量和自定义变量，可以理解为全局变量和局部变量，在理解他们的区别前，需要知道副程序和子程序，举个例子，当前的bash我们称之为父程序，而在这个bash下执行的任何程序，都称之为子程序。那全局变量和局部变量的区别就是，全局变量在子程序中仍然有效，局部变量只在当前程序中生效。（注意，一旦退出父程序这个bash的话，无论是全局变量还是局部变量都失效了，再次打开bash时，该变量是不存在的）

变量范围

**变量的设定**

变量的设定有几个规则，这里列举如下:  
- 设定变量通过=来链接，例如name=sujinzhou,两边不能有空格，如果有空格的话，需要用引号  
- 变量名只能是英文名和数字，只能是英文开头  
- 如果变量内容被引号扩起来的话，双引号和单引号是有区别的  
- 双引号里面的特殊字符，会保持原有的特性。  
- 单引号里面的特殊字符，只会是字符  
- 使用\将特殊字符转换为一般字符  
- 使用“这个符号扩起来的命令会先计算这个命令里面的内容，这个指令等价于$()  
- 给原有的变量添加值时，使用变量,"变量,"变量”再加具体的内容

变量类型

bash中变量类型有两种，一种是普通的字符串，一种是整数;设置变量的类型使用的是declare命令，delcare的命令格式如下

declare -[axir] variable

参数的意思:  
- -a:将variable定义为数组  
- -i:将后面的变量设置为整数类型  
- -x:与export一样，将后面的变量设置为环境变量  
- -r:将后面的变量设置为只读变量  
如果declare命令没有接任何变量的话，则表示显示所哟的

定义数组

var[index]=content

**变量的删除**

删除变量使用如下命令  
- unset 变量名

**变量的查看**

查看单个命令的内容，可以使用echo命令，使用echo $变量名就可以查看变量内容，另外如果要查看所有变量有3个命令，分别是如下几个命令，每个命令又有区别。  
- **echo $变量名** ;用于显示制定变量  
- **env**  ;用来查询环境变量，  
- **set**  ;显示shell变量，包括私有变量与用户变量，不同的shell有不同的私有变量，在shell中设置变量之后，可以通过set查到  
- **export**  ;显示当前导出成用户变量的shell变量，也是用来查询环境变量的，shell的私有变量不会出现

**Bash shell的操作环境**

在shell中设置的变量，无论是环境变量还是自定义变量，在bash中注销登录之后，就失效了。下次登录时，是没有这些变量的，如果想要这些变量永久生效，是需要下如配置文件的，bash在启动时，会读取这个配置文件，来设置当前shell的各种变量。shell分为两种，分别是login shell和nonlogin shell，这里先说login shell；login shell是需要输入用户名和密码的，一般情况下，login shell在登录之后，会去读取两个文件，来设置当前shell的环境变量，分别是  
1. /etc/profile:这个配置文件是默认配置文件，是系统整体的设置  
2. ~/.bash\_profile或者~/.bash\_login或者~/.profile：这个是用户的自定义设置文件

shell在读取/etc/profile中的文件之后，会去读取用户自定义的设置文件，按照顺序读取上文中所说的自定义设置文件，只要找到其中任何一个，就不再继续。在用户自定义的配置文件中，都会调用~/.bashrc这个文件，来配置当前的这个shell

/etc/profile或者~/.bash\_profile都是在登录的时候才去读取，如果将自己的一些设置到这些文件之后，一般都需要重新登录一次shell才会生效，也可以使用source这个命令，来读取配置文件的内容到当前的这个shell

**所以Linux下环境变量设置的三种方法**：

如想将一个路径加入到$PATH中，可以像下面这样做：

**只对当前的shell 起作用的环境变量**

### **1、**针对当前会话生效 【重启失效】

控制台中设置，不赞成这种方式，因为他只对当前的shell 起作用，换一个shell设置就无效了：

直接控制台中输入 ： $PATH="$PATH":/NEW\_PATH  (关闭shell Path会还原为原来的path)

**对所有的用户的都起作用的环境变量**

### **2、**/etc/profile 【永久有效】

修改 /etc/profile 文件，如果你的计算机仅仅作为开发使用时推存使用这种方法，因为所有用户的shell都有权使用这个环境变量，可能会给系统带来安全性问题。这里是针对所有的用户的，所有的shell

vi /etc/profile

在/etc/profile的最下面添加：  export  PATH="$PATH:/NEW\_PATH"

**针对当前特定的用户起作用的环境变量**

**3、**修改bashrc文件，这种方法更为安全，它可以把使用这些环境变量的权限控制到用户级别，这里是针对某一特定的用户，如果你需要给某个用户权限使用这些环境变量，你只需要修改其个人用户主目录下的 .bashrc文件就可以了。

vi ~/.bashrc

在下面添加：

Export  PATH="$PATH:/NEW\_PATH"

php,mysql的永久方法跟这个一样

**4、 使配置立即生效**

**source /etc/profile**

下来配置环境变量

在/etc/profile 中加入：

export NGINX\_HOME=/usr/local/nginx  
export PATH=$PATH:$NGINX\_HOME/sbin

和这种形式一样： export PATH=$PATH:/usr/local/nginx/sbin

保存，

执行 source /etc/profile ，使配置文件生效。