



北京邮电大学

Beijing University of Posts and Telecommunications



# 函数

蒋砚军 北京邮电大学计算机学院

## ► shell函数



### ■ 语法

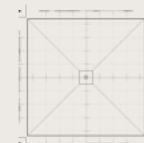
*name()* { *list*; }

### ■ 参数引用

- ◆ 函数定义完成之后，该函数名作为一个自定义内部命令执行，后面可以调用
- ◆ **调用时**函数名后附加上0到多个参数
- ◆ 在**函数体内部**以\$1，\$2，...或\$\*，\$@方式引用

### ■ 返回值

- ◆ 函数体内用内部命令return使函数有返回码，0表示成功，非零表示失败
- ◆ 函数内部可以创建和修改变量，函数返回后其它程序可以访问



## ► shell函数举例

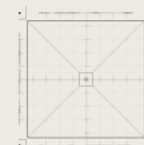


给出用户提示信息和默认值，等待用户输入配置信息。

要求：用户的输入必须从一个列表中选择。

用户直接按下回车则选择默认值，否则对输入值进行检查，强行用户在列表中选择。

配置结束后，将配置信息赋值给某一指定名称的变量，后续的程序中使用这些变量。



## ► 主程序



```
# main()
```

# 下面的程序中使用了“续行”，注意：反斜线后面不可以有任何空格或其他字符

```
get_attr COLOR "Color of the box" white \  
    "red green pink white black"
```

```
get_attr DAY "Day of the week" mon \  
    "sun mon tue wed thu fri sat"
```

```
get_attr ROLE "Your role in the system" student \  
    "admin student teacher guest"
```

```
# do system configuration here  
echo "COLOR=$COLOR DAY=$DAY ROLE=$ROLE"
```

## ► 函数定义（要放在调用程序之前）



# Usage: get\_attr 变量名 提示信息 默认值 取值表

```
get_attr()
```

```
{
```

```
    while true
```

```
    do
```

```
        echo -e "$2 [$3] : \c"; read val
```

```
        [ "$val" = "" ] && val=$3
```

```
        for i in $4
```

```
        do
```

```
            [ "$val" = "$i" ] && break 2
```

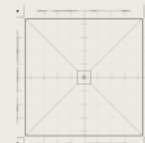
```
        done
```

```
        echo -e "**** Invalid choice \"$val\", must be in \"$4\""
```

```
    done
```

```
    eval "$1=$val"
```

```
}
```





北京邮电大学

Beijing University of Posts and Telecommunications



谢谢