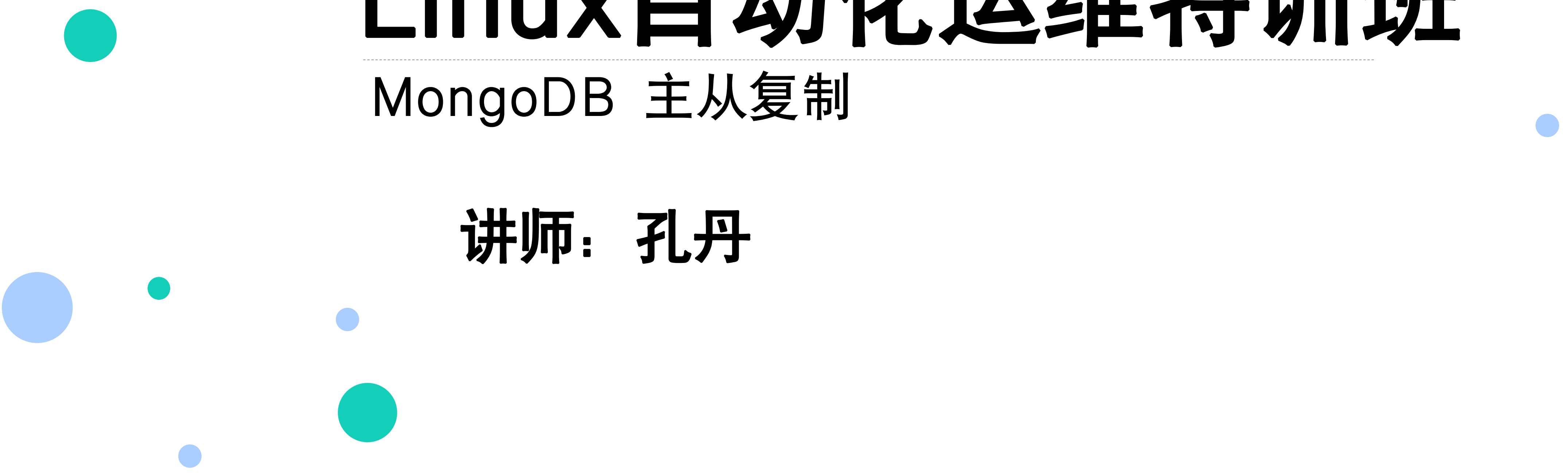




# Linux自动化运维特训班

脚本调试

讲师：孔丹



# 大纲

- Shell脚本中的常见错误：主要介绍Shell程序设计中经常遇到的语法错误和逻辑错误。
- Shell脚本调试技术：主要介绍Shell脚本的调试方法，包括简单的echo命令，trap命令，tee命令以及钩子程序调试方法等。

# Shell脚本中的常见错误

- 实际上，编写Shell脚本的过程就是不断排除错误的过程。尤其对于初学者来说，由于不熟悉Shell脚本的语法，会经常出现一些意想不到的错误。本节列出了Shell程序设计中经常出现的错误，从而使得读者在编程时可以避免这些错误的发生。

# 常见语法错误

□ 在进行Shell程序设计的时候，语法错误经常出现在关键字书写错误，引号错误，漏掉空格符以及变量的大小写问题等。

// if 条件语句缺少结尾关键字引起的错误

```
cat 10_1.sh
```

```
#!/bin/sh
```

```
if [ 10 -lt 12 ]
```

```
then
```

```
echo "Yes,10 is less than 12"
```

```
# sh 10_1.sh
```

```
10_1.sh:行5: 语法错误: 未预期的文件结尾
```

# 常见语法错误

□ 在进行Shell程序设计的时候，语法错误经常出现在关键字书写错误，引号错误，漏掉空格符以及变量的大小写问题等。

// 循环结构语句中缺少关键字引起的错误

for,while、 imtil 和 case 语句中的错误是指实际语句段不正确，也许是漏写或拼错了固定结构中的一个保留字。

```
# cat 10_2.sh
```

```
#!/bin/bash
```

```
n=1
```

```
while [ $n -le 5 ]
```

```
    echo "the numberis $n."
```

```
    let "n+=1"
```

```
done
```

```
# sh 10_2.sh
```

```
10_2.sh:行7: 未预期的符号 `done' 附近有语法错误
```

```
10_2.sh:行7: `done'
```

# 常见语法错误

// 成对的符号有 []、()、{}、""、"、``等， 如果它们落了单， 也会导致一些错误

```
[root@localhost ~]# cat 10_3.sh  
#!/bin/bash
```

```
while :  
do  
    read x  
    if [ $x == "exit"]; then  
        exit 0  
    else  
        echo "$x"  
    fi  
done
```

```
[root@localhost ~]# sh 10_3.sh
```

3

10\_3.sh: 第 6 行[: 缺少 `']

3

# 常见逻辑错误

- 通常情况下，Shell脚本中的语法错误是非常明显的，并且语法错误一般会导致程序不可执行。但是逻辑错误却比较隐蔽，因为这些错误通常不会引起程序执行失败，但是逻辑错误却会导致程序得到错误的结果。因此，相比语法错误，逻辑错误调试起来会更加困难。

```
[root@localhost ~]# cat 10_4.sh
```

```
#!/bin/bash
```

```
x=1
```

```
if [ x == 1 ]
```

```
then
```

```
    echo "x=1"
```

```
elif [ x == 0 ]
```

```
then
```

```
    echo "x=0"
```

```
else
```

```
    echo "other"
```

```
fi
```

```
[root@localhost ~]# sh 10_4.sh
```

```
other
```



# Shell脚本调试技术

□ 任何一个Shell开发者，都必须掌握所需要的调试技术，对于初学者来说，这显得尤为重要。因为在编写Shell脚本的时候，经常会遇到各种各样的错误，通过调试技术，可以快速排除错误。本节将介绍Shell编程中经常使用的几种调试技术。

1、使用dos2unix命令处理在windows下开发的脚本

2、使用echo命令调试

echo命令是最有用的调试脚本的工具之一。一般应在可能出现的问题的脚本重要的部分加入echo命令（在变量读取或修改操作的前后加入echo命令，并紧挨着退出命令exit）。此调试方法不仅适用于shell，在php、Python语言中经常使用。

3、使用bash命令参数调试

```
[root@ks ~]# sh [-nvx] nginx.sh
```

参数说明：

-n:不会执行该脚本，仅查询脚本语法是否有问题，并给出错误提示

-v:在执行脚本时，先将脚本的内容输出到屏幕上，然后执行脚本。如果有错误，也会给出错误提示。

-x:将执行的脚本内容及输出显示到屏幕上，这是对调试很有用的参数。



# Shell脚本调试技术

□ 任何一个Shell开发者，都必须掌握所需要的调试技术，对于初学者来说，这显得尤为重要。因为在编写Shell脚本的时候，经常会遇到各种各样的错误，通过调试技术，可以快速地排除错误。本节将介绍Shell编程中经常使用的几种调试技术。

## 4、使用set命令调试部分脚本内容

set 命令也可以用于辅助脚本调试。

set命令常用的调试选项：

set -n :读取命令单并不执行

set -v :显示读取的所有行

set -x :显示所有命令及其参数

提示：通过set -x 命令开启调试功能，而通过set +x 关闭调试功能。

set命令最大优点是，和bash -x相比，set -x可以缩小调试的作用域。

# Shell脚本调试技术

```
[root@ks ~]# cat debug.sh
#!/bin/bash
read -p "please input two number:" a b
set -x      #开启调试
if (($a < $b))
then
    echo "$a<$b"
elif (($a == $b))
then
    echo "$a=$b"
else
    echo "$a>$b"
fi
set +x      #结束调试，只针对set -x 和set +x 之间的脚本进行调试

echo "完美"
```

# 总结

- shell语法错误
- shell逻辑错误
- shell调试技巧

# 作业

- 写一个shell脚本，检查指定的shell脚本是否有语法错误，若有错误，首先显示错误信息，然后提示用户输入q或者Q退出脚本，输入其他内容则直接用vim打开该shell脚本。



# 谢谢观看

更多好课，请关注[万门大学APP](#)

