

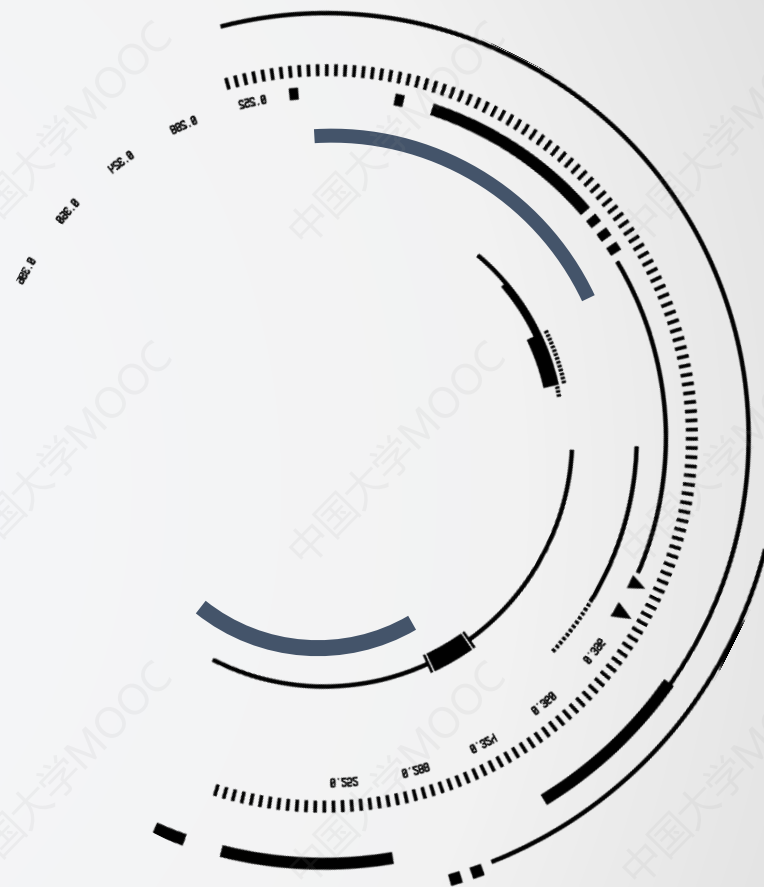
循环与循环控制

主讲教师：龙霄汉



◆ shell中的重复工作：**循环**

- **for**循环
- **while**循环
- **until**循环



for循环



常州信息职业技术学院

for 变量 **in** 可迭代的内容

do

循环体

done

变量 (**依次**) = 可迭代的内容

执行

循环体

可迭代内容全部遍历完

结束

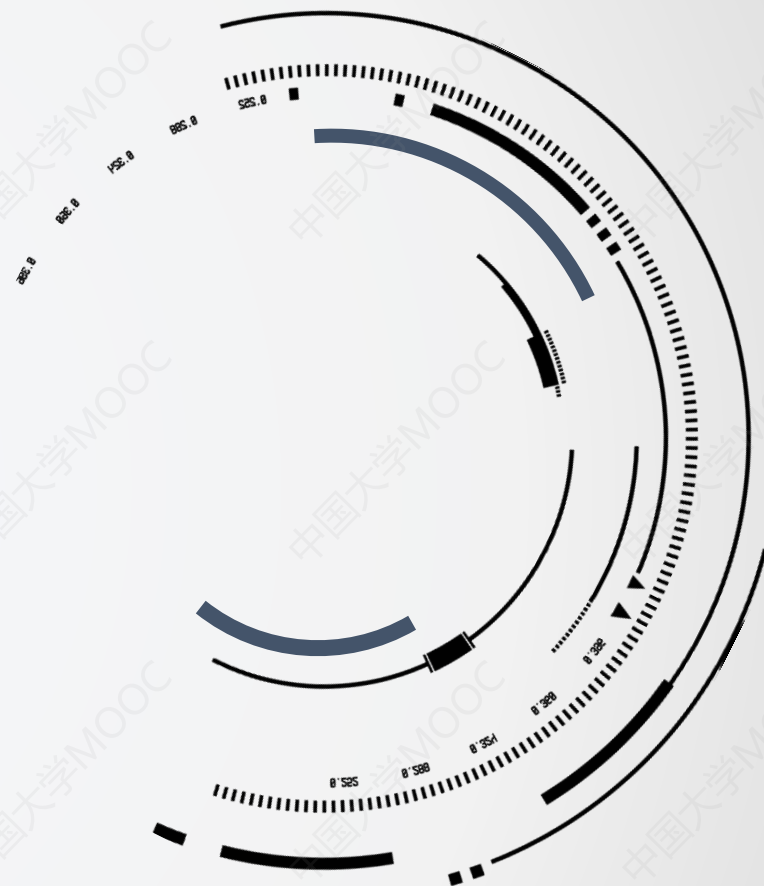


for循环格式



常州信息职业技术学院

for 变量 **in** 可迭代的内容 \longrightarrow **数字型**
do **{a..b}型**
循环体 **指令型**
done



操作演示



常州信息职业技术学院

Linux基础



◆ {a..b}型:

- **for i in** {a..b}

- 循环过程中, i的值表示从a 到b, 每次增加1

◆ 例如: for i in {1..10}

- 循环过程中, i的值为1、2、3 ... 10



操作演示



常州信息职业技术学院

Linux基础



◆ 指令型：

- for i in **指令**
- 循环过程中，i的值是指令的结果
- 每次循环之间按照**空白字符**（如空格、换行、TAB）作为**分隔符**



操作演示



常州信息职业技术学院

Linux基础

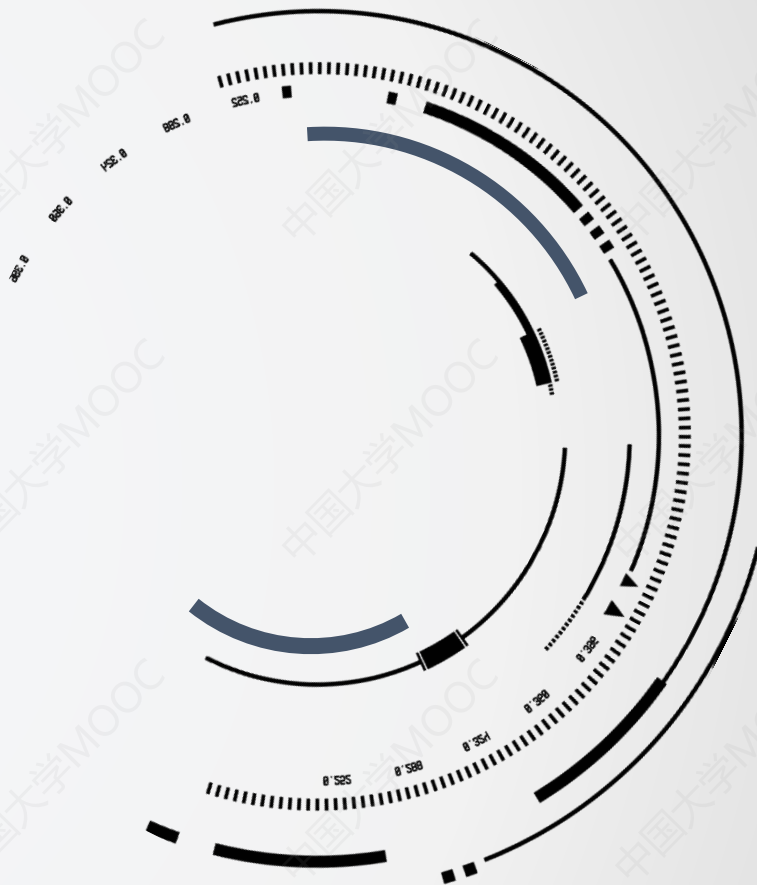


while循环



常州信息职业技术学院

- ◆ 循环次数**确定**时，用for 循环。
- ◆ 循环次数**不确定**时，用while 循环或者until 循环。



操作演示



常州信息职业技术学院

Linux基础



- ◆ until循环与while循环正好相反，当**条件不成立时，执行循环**，直到条件成立。



```
#!/bin/bash
i=0
while [ $i -lt 10 ]
do
    let i=i+1
    echo $i
done
```



◆ 在循环的过程中，可以使用**continue**和 **break**进行**循环控制**：

- **continue**表示退出当前循环，开始下一次循环
- **break**表示跳出整个循环，不再进行循环



感谢您的观看!

