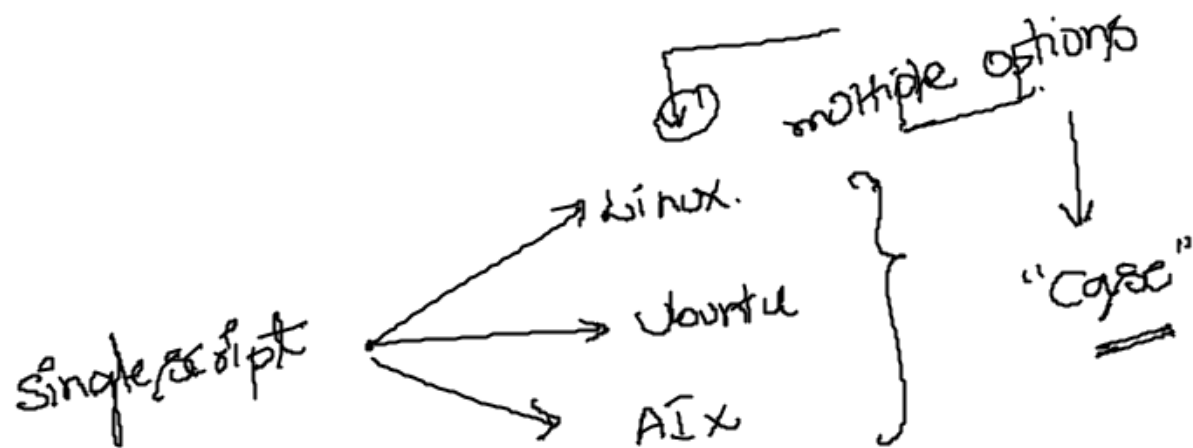


shell scripting

loops



Syntax

```

case $1 in
  "val1") stmt1 ;
            stmt2 ;;
  "val2") stmt3 ;;
  ".")
    echo "usage" ;;
esac
  
```

Select

os related work as a service



① F.S

② user

③ pkg

④ OS Manager

Special use cases

Deep

break ÷

uptime

⑦

↓
Every 10 seconds

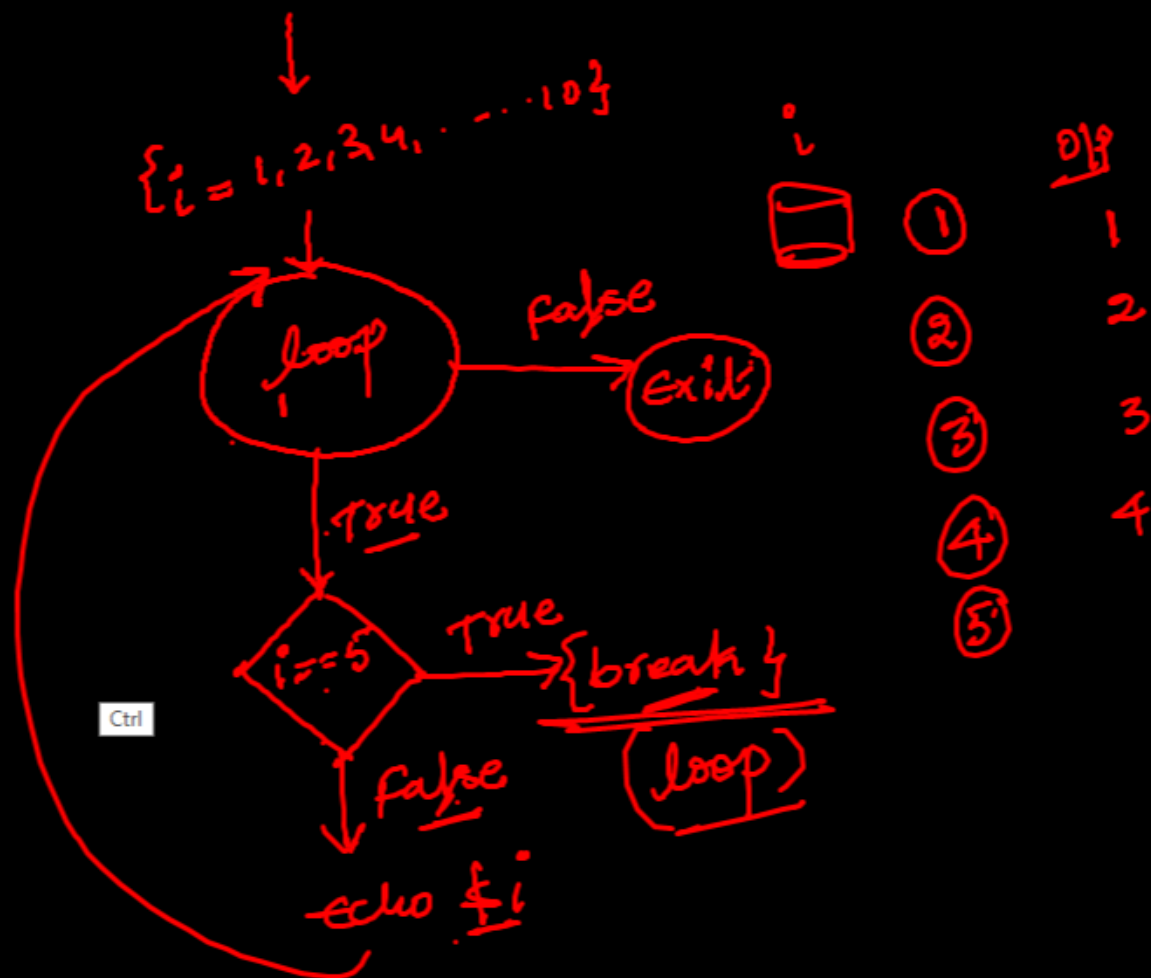
```
[root@shellscripting day2]# cat mybreak.sh
#!/bin/bash
```

```
for i in {1..10}
do
    if (( i == 5 ))
    then
        break
    fi
    echo $i
done
```

```
[root@shellscripting day2]# bash mybreak.sh
```

```
1
2
3
4
```

```
[root@shellscripting day2]#
```



```
[root@shellscripting day2]# vi mybreak.sh
[root@shellscripting day2]# cat mybreak.sh
#!/bin/bash
```

```
for i in {1..10}
do
    if (( i == 5 ))
    then
        (break)
    fi
    echo $i
done
```

```
echo "My second loop"
for j in a b c d e f
do
    echo $j
done
```

```
[root@shellscripting day2]# bash mybreak.sh
```

```
1
2
3
4
My second loop
a
b
c
d
e
f
```

```
[root@shellscripting day2]#
```

o/p

1
2
3
4
a
b
c
d
e
f

break will
come out of loop only

2nd

Collapse

```
[root@shellscripting day2]# cat mybreak.sh
```

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

```
for i in {1..10}
```

```
do
```

```
  if (( i == 5 ))
```

```
  then
```

```
    exit
```

```
  fi
```

```
    echo $i
```

```
done
```

} 1st block

exit will come out of
entire code & it will not
allow others to work

```
echo "My second loop"
```

```
for j in a b c d e f
```

```
do
```

```
  echo $j
```

```
done
```

} 2nd block

```
[root@shellscripting day2]# bash mybreak.sh
```

```
1
```

```
2
```

```
3
```

```
4
```

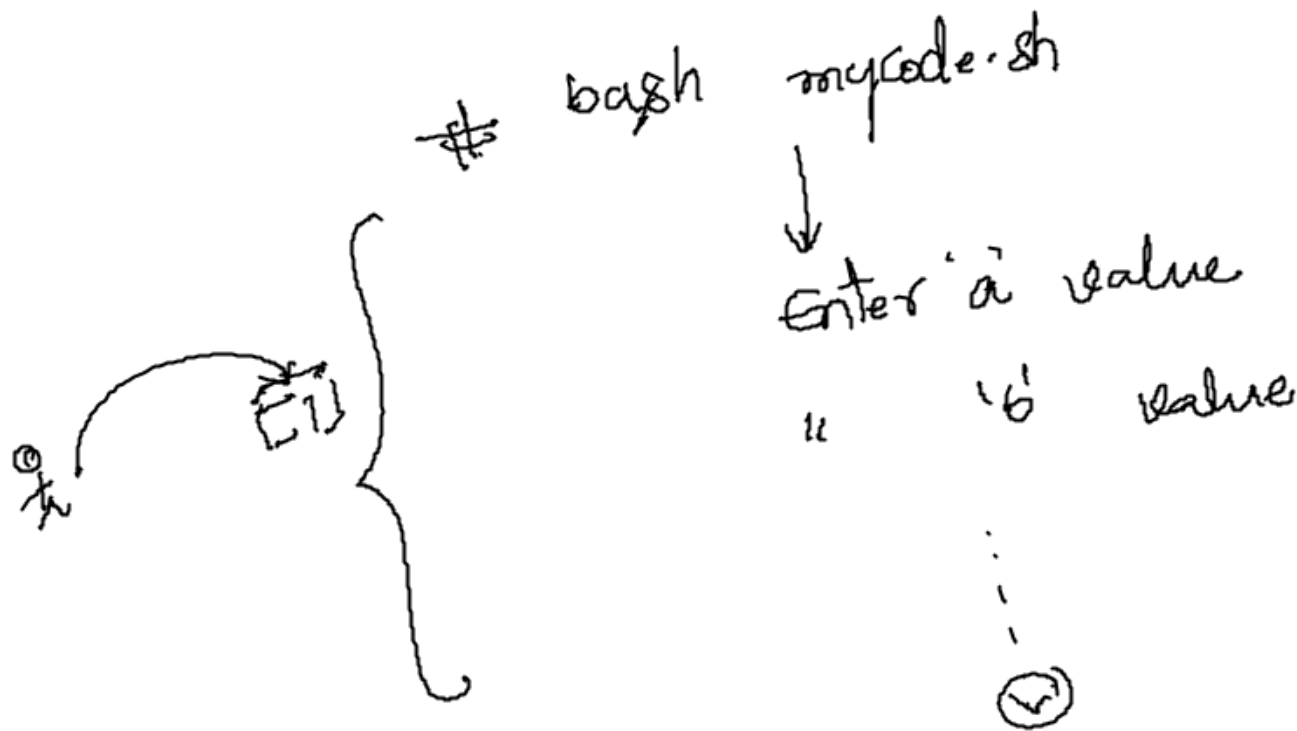
```
[root@shellscripting day2]#
```

Command Line Argument's



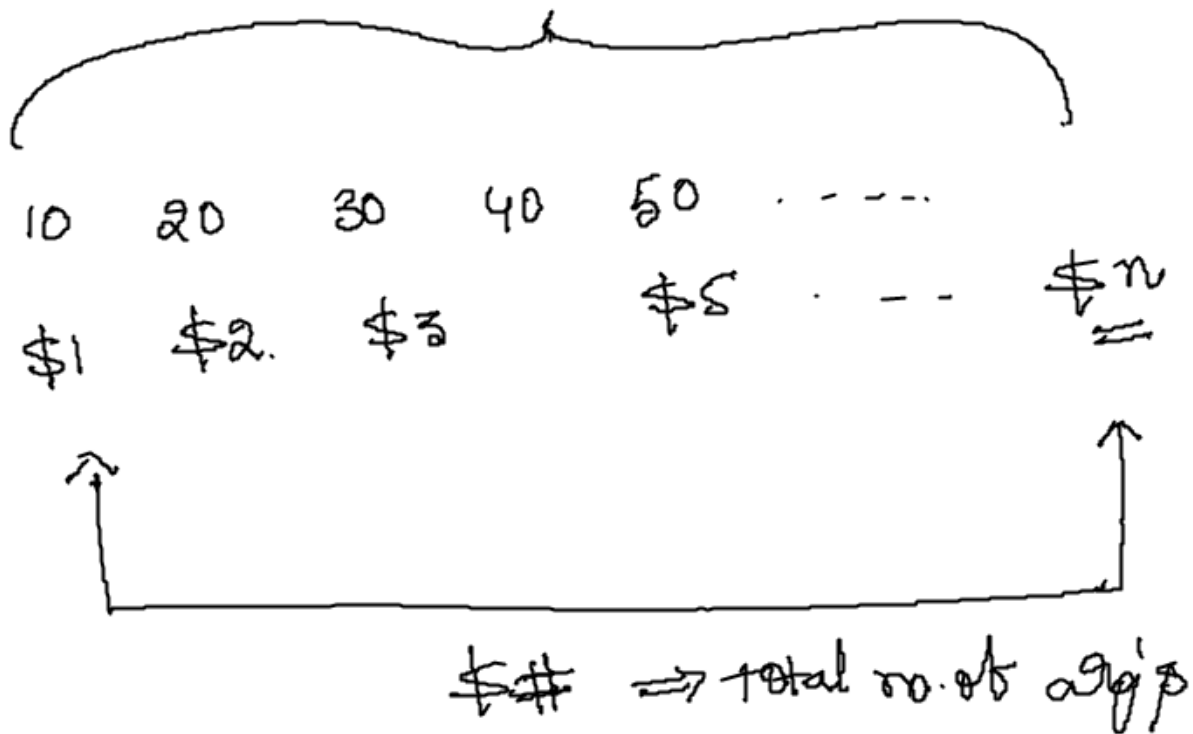
argument == parameter == value to code

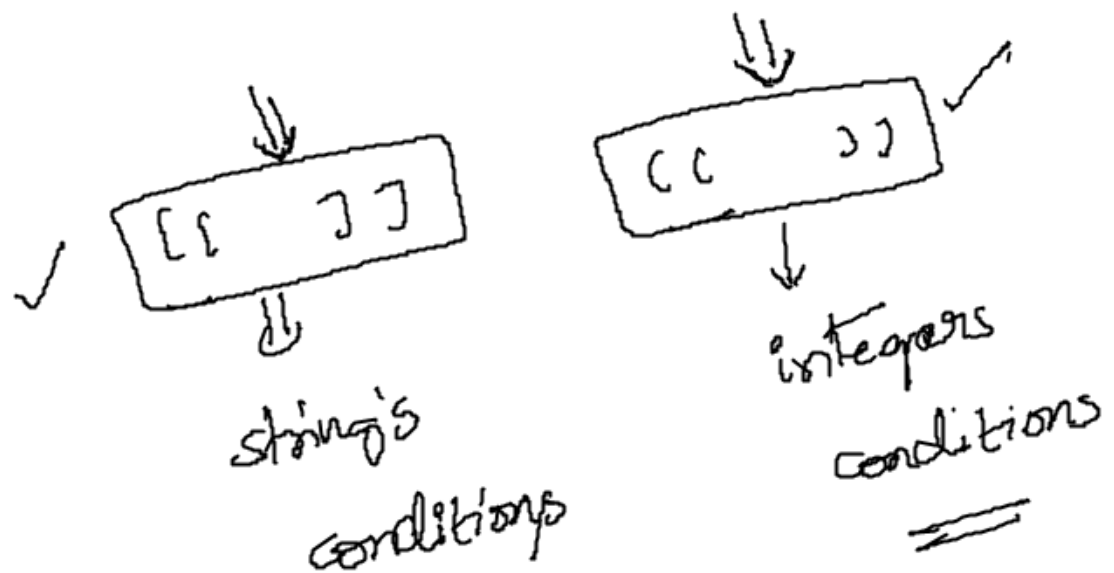
7/11

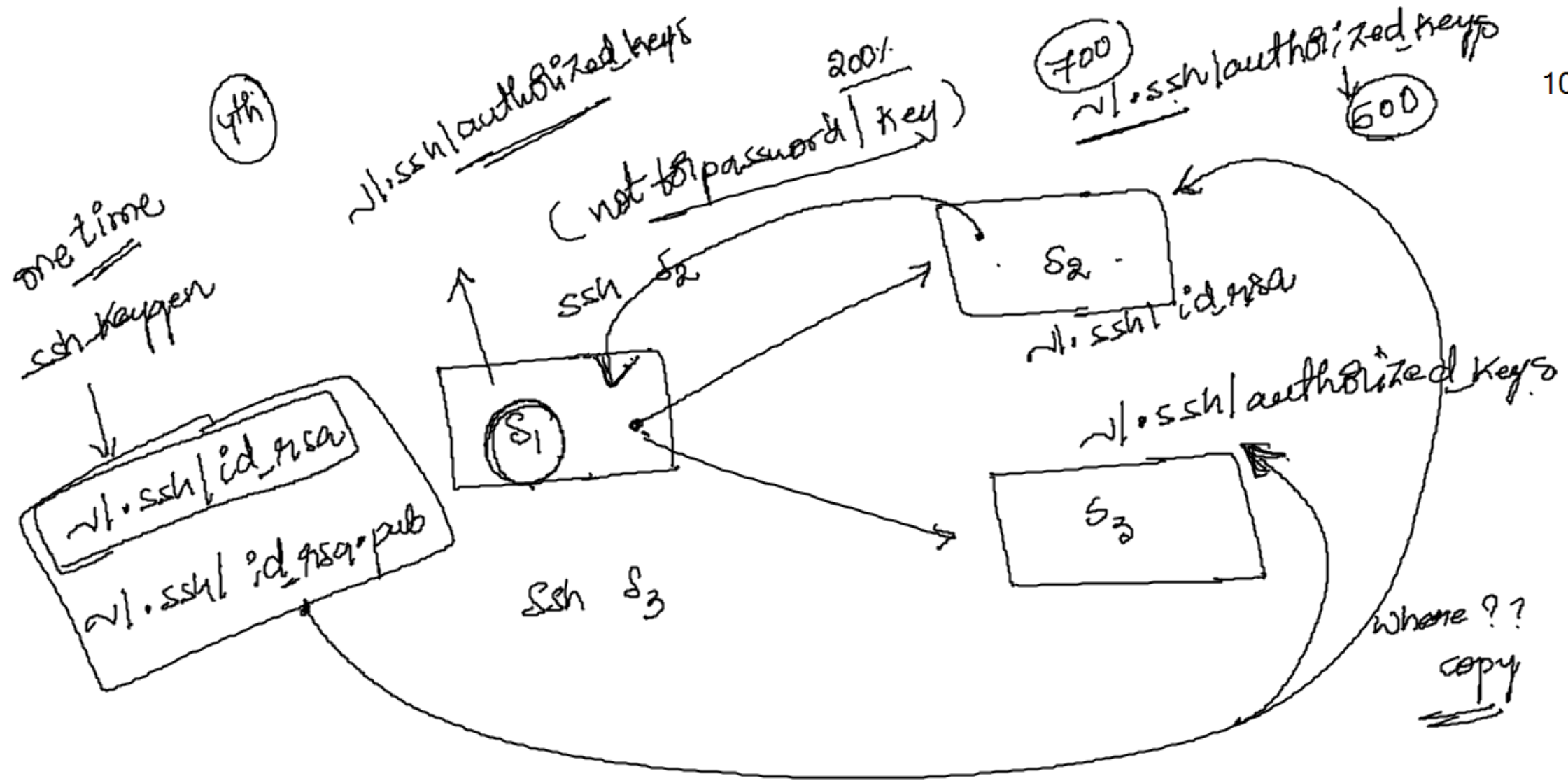


bash myCode.sh
\$0

"\$*" holds all values







cc

[c]

execution

$\$(\$a)$

cca