

LINUX CONDITIONAL STATEMENTS

Conditional Statements

-eq : equal	= : equal	-d file : directory
-ne : not equal	! : not equal	-e file : exists
-gt : greater than	-z : empty string	-f file : ordinary file
-lt : less than	-n : not empty string	-r file : readable
-ge : greater than or equal	! : negation	-w file : writable
-le : less than or equal	&& : and	-x file : executable
	: or	-s file : size is > 0 bytes

IF Statements

```
#!/bin/bash
read -p "Input a number: " number

if [[ $number -gt 50 ]]
then
    echo "The number is big."
fi
```

Relational Operators - String Operators

```
#!/bin/bash

if [[ "a" = "a" ]]
then
    echo "They are same"
fi

if [[ "a" != "b" ]]
then
    echo "They are not same"
fi

if [[ -z "" ]]
then
    echo "It is empty"
fi

if [[ -n "text" ]]
then
    echo "It is not empty"
fi
```

File Test Operators

```
#!/bin/bash

if [[ -d folder ]]
then
    echo "folder is a directory"
fi

if [[ -f file ]]
then
    echo "file is an ordinary file"
fi

if [[ -r file ]]
then
    echo "file is a readable file"
fi

if [[ -w file ]]
then
    echo "file is a writable file"
fi

if [[ -s file ]]
then
    echo "file is > 0 bytes"
fi

if [[ -x $0 ]]
then
    echo "$0 is an executable file "
fi
```

IF ELSE Stataments

```
#!/bin/bash
read -p "Input a number: " number

if [[ $number -ge 10 ]]
then
    echo "The number is bigger than or equal to 10."
else
    echo "The number is smaller than 10"
fi
```

IF ELIF ELSE Statements

```
#!/bin/bash
read -p "Input a number: " number

if [[ $number -eq 10 ]]
then
    echo "The number is equal to 10."
elif [[ $number -gt 10 ]]
then
    echo "The number is bigger than 10"
else
    echo "The number is smaller than 10"
fi
```

Nested IF Statements

```
#!/bin/bash
read -p "Input a number: " number

if [[ $number -gt 10 ]]
then
    echo "Number is bigger than 10"

    if (( $number % 2 == 1 ))
    then
        echo "And is an odd number."
    else
        echo "And is an even number."
    fi
else
    echo "It is not bigger than 10"
fi
```

Boolean Statements

```
read -p "Input your name: " name
read -sp "Input your password: " password

if [[ $name = $(whoami) ]] && [[ $password = Aa1234 ]]
then
    echo -e "\nWelcome $(whoami)"
else
    echo -e "\nIt is wrong account"
fi
```

Case Statements

```
#!/bin/bash

read -p "Input first number: " first_number
read -p "Input second number: " second_number
read -p "Select an math operation
1 - addition
2 - subtraction
3 - multiplication
4 - division
" operation

case $operation in
    "1")
        echo "result= $(( $first_number + $second_number))"
        ;;
    "2")
        echo "result= $(( $first_number - $second_number))"
        ;;
    "3")
        echo "result= $(( $first_number * $second_number))"
        ;;
    "4")
        echo "result= $(( $first_number / $second_number))"
        ;;
    *)
        echo "Wrong choice..."
        ;;
esac
```

Loop Statements

While Loops

For Loops

Select Loops

Until Loops

For Arrays

Continue Statements

Break Statements

While Loops (içerisi doğru iken do-ya girer)

```
#!/bin/bash

number=1

while [[ $number -le 10 ]]
do
    echo $number
    ((number++))
done
echo "Now, number is $number"
```

Until Loops (içerisi yanlış iken do-ya girer)

```
#!/bin/bash

number=1

until [[ $number -ge 10 ]]
do
    echo $number
    ((number++))
done
echo "Now, number is $number"
```

For Loops

```
#!/bin/bash

echo "Numbers:"

for number in {0..9}
do
    echo $number
done

echo "Names:"
for name in Tadic Icardi Aboubakar Bakasetas Wright
do
    echo $name
done

echo "Files in current folder:"
for file in `pwd`/*
do
    echo $file
done
```

For Arrays

```
#!/bin/bash

devops_tools=("docker" "kubernetes" "ansible" "terraform" "jenkins")

for tool in ${devops_tools[@]}
do
    echo $tool
done
```

(@ işareti bütün elemanları çağır demek)

Break Statements

```
#!/bin/bash

number=1

until [[ $number -lt 1 ]]
do
    echo $number
    ((number++))
    if [[ $number -eq 100 ]]
    then
        break
    fi
done
echo "Now, number is $number"
```

Continue Statements

```
#!/bin/bash

number=1

until [[ $number -lt 1 ]]
do
    ((number++))

    tens=$(( $number % 10 ))

    if [[ $tens -eq 0 ]]
    then
        continue
    fi

    echo $number

    if [[ $number -gt 100 ]]
    then
        break
    fi
done
```

Select Loops

```
#!/bin/bash

read -p "Input first number: " first_number
read -p "Input second number: " second_number

PS3="Select the operation: "

select operation in addition subtraction multiplication division exit
do
    case $operation in
        addition)
            echo "result= $(( $first_number + $second_number))"
            ;;
        subtraction)
            echo "result= $(( $first_number - $second_number))"
            ;;
        multiplication)
            echo "result= $(( $first_number * $second_number))"
            ;;
        division)
            echo "result= $(( $first_number / $second_number))"
            ;;
        exit)
            break
            ;;
        *)
            echo "Wrong choice..."
            ;;
    esac
done
```

```
#!/bin/bash

read -p "Input first number: " first_number
read -p "Input second number: " second_number

PS3="Select the operation: "

select operation in addition subtraction multiplication division exit
do
    case $operation in
        addition)
            echo "result= $(( $first_number + $second_number))"
            ;;
        subtraction)
            echo "result= $(( $first_number - $second_number))"
            ;;
        multiplication)
            echo "result= $(( $first_number * $second_number))"
            ;;
        division)
            echo "result= $(( $first_number / $second_number))"
            ;;
        exit)
            break
            ;;
        *)
            echo "Wrong choice..."
            ;;
    esac
done
```

```
deniz@DESKTOP-F7L0BJN:~/loops$ nano select-loop.sh
deniz@DESKTOP-F7L0BJN:~/loops$ ./select-loop.sh
Input first number: 100
Input second number: 50
1) addition
2) subtraction
3) multiplication
4) division
5) exit
Select the operation: 1
result= 150
Select the operation: 55
Wrong choice...
Select the operation: 5
deniz@DESKTOP-F7L0BJN:~/loops$
```

Functions

Script

```
#!/bin/bash

Welcome () {
    echo "Welcome to the Linux Lessons"
}

Welcome
```

Fonksiyonun ismini yazarak fonksiyonu çalıştırıyoruz.

Terminalde de tanımlanıp çağırılabilir.

```
oguzhan@DESKTOP-C9CE1DQ:~/functions$ Necip () {
    echo "Welcome to Linux Lessons"
}
oguzhan@DESKTOP-C9CE1DQ:~/functions$ Necip
Welcome to Linux Lessons
```

Parametre kullanımı

```
#!/bin/bash

Welcome () {
    echo "Welcome to the Linux Lessons $1 $2 $3"
}

Welcome Begüm Umut Baha
```

```
deniz@DESKTOP-F7LOBJN:~/functions$ cat functions.sh
#!/bin/bash

Welcome () {
    echo "Welcome to Linux Lessons $1 $2 $3"
    return 3
}

Welcome Elif Necip Begum
echo $?
deniz@DESKTOP-F7LOBJN:~/functions$ ./functions.sh
Welcome to Linux Lessons Elif Necip Begum
3
deniz@DESKTOP-F7LOBJN:~/functions$
```

```
deniz@DESKTOP-F7LOBJN:~/functions$ cat functions.sh
#!/bin/bash

Welcome () {
    echo "Welcome to Linux Lessons $1 $2 $3"
    return 3
}

Welcome Elif Necip Begum
pwd
echo $?
deniz@DESKTOP-F7LOBJN:~/functions$ ./functions.sh
Welcome to Linux Lessons Elif Necip Begum
/home/deniz/functions
0
deniz@DESKTOP-F7LOBJN:~/functions$
```

Echo son dönen değeri return etti. İlkinde bu yüzden direk fonksiyon çıktısını aldı.

Ama sonrakinde fonksiyon sonrasında pwd komutu çağırıldığı için, sondaki echo komut kontrolünden 0 çıktısı geldi.

Nested Functions

```
#!/bin/bash

function_one () {
    echo "This is from function_one"
    function_two
}

function_two () {
    echo "This is from function_two"
}

function_one
```

```
deniz@DESKTOP-F7LOBJN:~/functions$ cat nested-functions.sh
#!/bin/bash

function_one () {
    echo "This is from function_one"
    function_two
}

function_two () {
    echo "This is from function_two"
}

function_one

deniz@DESKTOP-F7LOBJN:~/functions$ ./nested-functions.sh
This is from function_one
This is from function_two
deniz@DESKTOP-F7LOBJN:~/functions$
```

Variable Scopes

```
#!/bin/bash

var1='global 1'
var2='global 2'

var_scope () {
    local var1='function 1'
    var2='function 2'
    echo -e "Inside function:\nvar1: $var1\nvar2: $var2"
}

echo -e "Before calling function:\nvar1: $var1\nvar2: $var2"

var_scope

echo -e "After calling function:\nvar1: $var1\nvar2: $var2"
```

```
deniz@DESKTOP-F7LOBJN:~/functions$ ./variable-scope.sh
Before calling function:
var1: global 1
var2: global 2
Inside function:
var1: function 1
var2: function 2
After calling function:
var1: global 1
var2: function 2
deniz@DESKTOP-F7LOBJN:~/functions$
```

-sed command

First occurrence yani ilk defa denk geldiği kelimeyi değiştiriyor. Asıl dosya değişmiyor, çıktı değişiyor.

```
deniz@DESKTOP-F7LOBJN:~/sed-awk-command$ cat sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
deniz@DESKTOP-F7LOBJN:~/sed-awk-command$ sed 's/linux/ubuntu/' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes ubuntu.
deniz@DESKTOP-F7LOBJN:~/sed-awk-command$
```

Eğer sayı belirtilirse belirtilen denkleme değeri değiştirir.

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ cat sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ sed 's/linux/ubuntu/' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes ubuntu.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ sed 's/linux/ubuntu/3' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes ubuntu.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ |
```

Eğer *i* belirtilirse (büyük küçük duyarlı)

sed 's/linux/ubuntu/i' sed.txt

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ cat sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ sed 's/linux/ubuntu/i' sed.txt
ubuntu is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like linux. Everyone likes linux.
ubuntu is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$
```

Eğer *g* belirtilirse (global yani her tarafta)

sed 's/linux/ubuntu/g' sed.txt

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ cat sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ sed 's/linux/ubuntu/g' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like ubuntu. Everyone likes ubuntu.
Linux is free. Linux is good. Linux is hope. Oguzhan likes ubuntu.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ |
```

Eğer *ig* belirtilirse (hem global ve hem büyük küçük duyarlı)

sed 's/linux/ubuntu/ig' sed.txt

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ cat sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ sed 's/linux/ubuntu/ig' sed.txt
ubuntu is an OS. ubuntu is life. ubuntu is a concept.
I like ubuntu. You like ubuntu. Everyone likes ubuntu.
ubuntu is free. ubuntu is good. ubuntu is hope. Oguzhan likes ubuntu.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$
```

Eğer başta **2** belirtilirse

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ cat sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like linux. You like linux. Everyone likes linux.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ sed '2 s/linux/ubuntu/ig' sed.txt
Linux is an OS. Linux is life. Linux is a concept.
I like ubuntu. You like ubuntu. Everyone likes ubuntu.
Linux is free. Linux is good. Linux is hope. Oguzhan likes linux.
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$
```

-awk command

awk '{print}' file.txt	=> bütün dosyayı yazdırıyor
awk '/This/ {print}' file.txt	=> This ile başlayan satırları yazdırıyor
awk '/Merhaba/ {print}' file.txt	=> Merhaba ile başlayan satırları yazdırıyor

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '{print}' awk.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '/This/ {print}' awk.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ nano awk.txt
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '/This/ {print}' awk.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '/Merhaba/ {print}' awk.txt
Merhaba
```


awk '{print \$0}' file.txt
awk '{print \$1}' file.txt
awk '{print \$2}' file.txt
awk '{print \$2, \$4}' file.txt

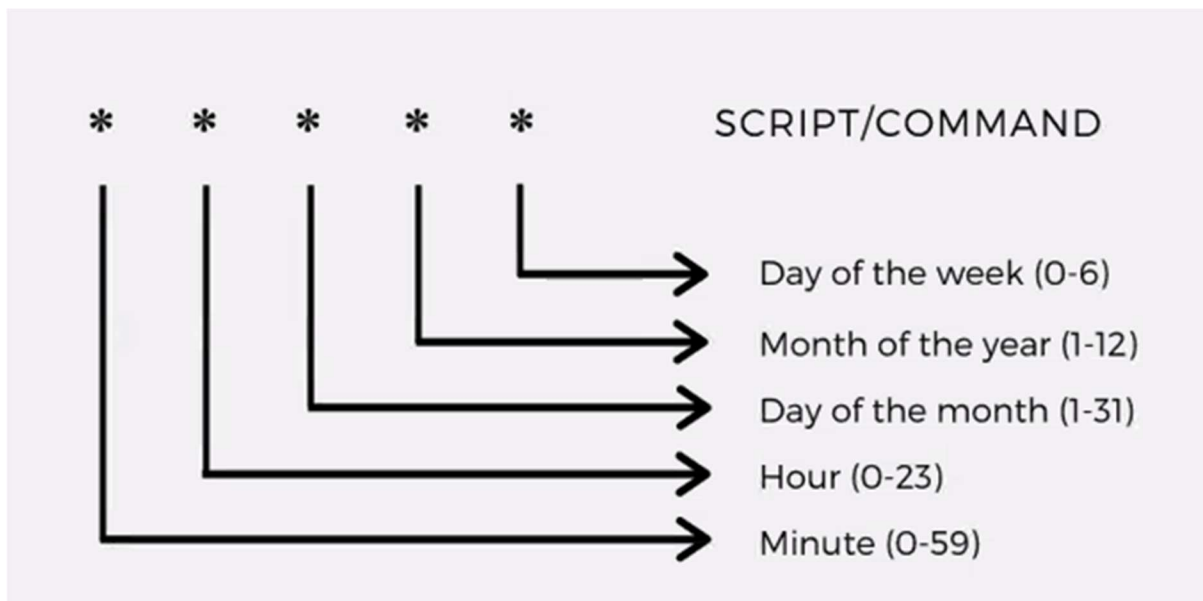
=> bütün dosyayı yazdırıyor
=> bütün dosya çıktısının birinci sütununu yazdırıyor
=> bütün dosya çıktısının ikinci sütununu yazdırıyor
=> bütün dosya çıktısının ikinci ve dördüncü sütununu yazdırıyor

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '{print $0}' awk.txt
This is line 1
This is line 2
This is line 3
This is line 4
This is line 5
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '{print $1}' awk.txt
This
This
This
This
This
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '{print $2}' awk.txt
is
is
is
is
is
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '{print $4}' awk.txt
1
2
3
4
5
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk '{print $2,$4}' awk.txt
is 1
is 2
is 3
is 4
is 5
```

awk -F: '{print \$2}' awk.txt => bütün dosya çıktısını F ile belirtilen karaktere göre ayırıp, sonra ayırma yapılır.

```
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ cat awk.txt
This is part 1 of line 1 : This is part 2 of line 1
This is part 1 of line 2 : This is part 2 of line 2
This is part 1 of line 3 : This is part 2 of line 3
This is part 1 of line 4 : This is part 2 of line 4
This is part 1 of line 5 : This is part 2 of line 5
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ awk -F: '{print $2}' awk.txt
This is part 2 of line 1
This is part 2 of line 2
This is part 2 of line 3
This is part 2 of line 4
This is part 2 of line 5
oguzhan@DESKTOP-C9CE1DQ:~/sed-awk-command$ nano |
```

-crontab command (belirli zaman aralıklarında tekrarlanan görevleri düzenlemek için kullanılır)



crontab -e yazarak dosyanın içine gidiyoruz ardından metin dosyasının altına otomatize etmek istediğimiz bir görevi yazıyoruz.

0 3 * * * username sudo apt update -y	#	Her gün saat 3.00-da sudo apt update görevi yapılır.
---------------------------------------	---	--

crontab -l yazarak güncel olan otomatik görevleri görüntülüyoruz.