BASH SCRIPT
---
1) Admin (yönetici) kişilerin işlerini kolaylaştırması,

2) Çalışanlara admin olmadığı zaman kolaylık sağlaması,

3) Ev kullanıcısı tüm kullanıcılara kolaylık sağlaması,

Bash script vi, vim, gedit, visual studio code vb. ortamlarda
yazılabilir.
---
--
```
#! /bin/bash
echo "Merhaba Arkadaslar"
```

Oluşturduğumuz script dosyasını executable yapıyoruz.
chmod +x dosya ismi

oluşturduğumuz dosyayı /usr/bin içerisine atarsanız her dizinden
çalıştırabilirsiniz.
---
--

DEĞİŞKENLER (VARIABLES)
---
1) System Variables
2) User Variables
---
--
READ KULLANIMI
---
```
echo "İsminiz: "
read isim
echo "İsmim $isim"

echo "İsimler: "
read isim1 isim2 isim3
echo "İsimler: $isim1, $isim2, $isim3"
```
---
```
read -p 'İsminiz: ' isim
read -sp 'Şifreniz: ' sifre
echo
echo "İsmim $isim"
echo "Şifrem $sifre"
```
---
--
ARGUMENT (ARGS)

```
#! /bin/bash

echo $0 $1 $2 $3 $4

echo $*
```

```
echo $#

dizi=("$@")
echo ${dizi[0]} ${dizi[3]}
-------------------------------------------------------------------------
--
IF KULLANIMI
------------
#! /bin/bash

if[Kosul]
then
   durum
fi

(if, elif, else)

INT KARŞILAŞTIRMA
-----------------
-eq / eşit ise             / if[ "$a" -eq "$b" ] / (equal)
-ne / eşit değil ise       / if[ "$a" -ne "$b" ] / (not equal)
-gt / büyük ise            / if[ "$a" -gt "$b" ] / (greater than)
-ge / büyük veya eşit ise  / if[ "$a" -ge "$b" ] / (greater than or
equal)
-lt / küçük ise            / if[ "$a" -lt "$b" ] / (less than)
-le / küçük veya eşit ise  / if[ "$a" -le "$b" ] / (less than or equal)

 <  / küçük      / if(("$a" <  "$b" ))
 <= / küçük eşit / (("$a" <= "$b" ))
 >  / büyük      / (("$a" >  "$b" ))
 >= / büyük eşit / (("$a" >= "$b" ))

STRING KARŞILAŞTIRMA
--------------------
 =  / eşit ise       / if[ "$a" =  "$b" ]
 == / eşit ise       / if[ "$a" == "$b" ]
 != / eşit değil ise / if[ "$a" != "$b" ]
 <  / küçük          / if[[ "$a" <  "$b" ]] / Alfabetik dizilime göre
 >  / küçük          / if[[ "$a" >  "$b" ]] / Alfabetik dizilime göre

AND ve OR Operatörleri
----------------------
AND --> && (-a)

#! /bin/bash

yas=32

if [ "$yas" -gt 18 ] && [ "$yas" -lt 30 ]
then
echo "Geçerli yaş"
else
echo "Geçersiz yaş"
fi
```

```
---------------------
OR  --> || (-o)

#! /bin/bash

yas=18

if [ "$yas" -eq 18 ] || [ "$yas" -lt 15 ]
then
echo "Geçerli yaş"
else
echo "Geçersiz yaş"
fi
-------------------------------------------------------------------------
--
ARİTMETİK İŞLEMLER

#! /bin/bash

sayi1=25
sayi2=5

#echo $(( sayi1+sayi2 ))
#echo $(( sayi2-sayi1 ))
#echo $(( sayi1*sayi2 ))
#echo $(( sayi1/sayi2 ))
#echo $(( sayi1%sayi2 ))

echo $( expr $sayi1 + $sayi2 )
echo $( expr $sayi1 - $sayi2 )
echo $( expr $sayi1 \* $sayi2 )
echo $( expr $sayi1 / $sayi2 )
echo $( expr $sayi1 % $sayi2 )

FLOAT SAYILAR
--------------
#! /bin/bash

sayi1=20.5
sayi2=5

echo "20.5+5" | bc
echo "20.5-5" | bc
echo "20.5*5" | bc
echo "20.5/5" | bc
echo "20.5%5" | bc

echo "scale=2;20.5/5" | bc

echo "scale=2;$sayi1/$sayi2" | bc
echo "$sayi1+$sayi2" | bc

echo "scale=10; sqrt($sayi2)" | bc -l
```

```
echo "scale=2; $sayi1^3" | bc -l
--------------------------------------------------------------------------
--
Dosya Doğrulama Operatörleri (File Check Operators)

-e dosya mevcut
-f dosya mevcut ve regular file
-s dosya içeriği dolu
-d klasör olup olmadığı
-r read
-w write
-x executable

#! /bin/bash

echo -e "Dosyanın ismini giriniz:\c"

read dosyaismi

if [ -e $dosyaismi ]
then
    echo "$dosyaismi bulundu"
else
    echo "$dosyaismi bulunamadı"
fi
-----------------------------------
#! /bin/bash

echo -e "Dosyanın ismini giriniz:\c"

read dosyaismi

if [ -s $dosyaismi ]
then
    echo "$dosyaismi içeriği dolu"
else
    echo "$dosyaismi boş"
fi
-----------------------------------
#! /bin/bash

echo -e "Dosyanın ismini giriniz:\c"

read dosyaismi

if [ -w $dosyaismi ]
then
    echo "$dosyaismi yazılabilir"
else
    echo "$dosyaismi yazılabilir değil"
fi
-----------------------------------
#! /bin/bash
```

```
echo -e "Dosyanın ismini giriniz:\c"

read dosyaismi

if [ -f $dosyaismi ]
then
    if [ -w $dosyaismi ]
    then
        echo "Dosya yazılabilir. Ctrl+d ile çıkabilirsiniz"
        cat >> $dosyaismi
    else
        echo "Dosya yazılabilir değil"
    fi
else
    echo "Dosya mevcut değil"
fi  # if kalıbını kapattık
-------------------------------------------------------------------------
--
CASE STATEMENT
--------------
#! /bin/bash

case değişken in
     değişken ifadesi)
     durum;;
     değişken ifadesi)
     durum;;
esac

-----------------------------------------------
#! /bin/bash
arac=$1

case $arac in
    "araba" )
    echo "$arac 200TL'ye günlük kiralanır";;
    "Motorsiklet" )
    echo "$arac 100TL'ye günlük kiralanır";;
    "Bisiklet" )
    echo "$arac 50TL'ye kiralanır";;
    * )
    echo "$arac yoktur";;
esac
-------------------------------------------------
#! /bin/bash

echo -e "Bir arac giriniz:\c"

read arac

case $arac in
    "araba" )
    echo "$arac 200TL'ye günlük kiralanır";;
    "Motorsiklet" )
```

```
    echo "$arac 100TL'ye günlük kiralanır";;
    "Bisiklet" )
    echo "$arac 50TL'ye kiralanır";;
    * )
    echo "$arac yoktur";;
esac
-------------------------------------------------
#! /bin/bash

echo -e "Bir Karakter Giriniz:\c"

read deger

case $deger in
    [a-z] )
    echo "Kullanıcı $deger girişi yaptı a-z arasında";;
    [0-9] )
    echo "Kullanıcı $deger girişi yaptı 0-9 arasında";;
    ? )
    echo "Kullanıcı $deger girişi yaptı özel karakter";;
    * )
    echo "Kullanıcı $deger girişi yaptı bilinmeyen";;
esac
-------------------------------------------------------------------------
--
DİZİLER
--------
#! /bin/bash

OS=( 'Linux' 'Windows' 'Unix' )

echo "${OS[@]}" #Tüm dizi elemanlerını gösterir

echo "${OS[2]}"

echo "${!OS[@]}" #Tüm dizinin index sırasını gösterir

echo "${#OS[@]}" #Tüm dizi eleman sayısını gösterir

OS[3]='Mac'

echo "${OS[@]}"

unset OS[1]

echo "${OS[@]}"

echo "${!OS[@]}"
-------------------------------------------------------------------------
--
WHİLE LOOPS
-----------
#! /bin/bash
```

```
i=1

while (($i<=2))
do
    echo $i
    ((i++))
    sleep 1
    gnome-terminal
done
--------------------------------------------------------------------------------
---
FOR LOOPS
---------
#! /bin/bash

for (( i=0;i<=5;i++ ))
do
  echo $i
done
--------
for i in ls pwd
do
  echo "----------$i-----------"
  $i
  echo
done
--------
#! /bin/bash

for i in {1..10}
do
echo $i
done
---------
#! /bin/bash

for i in {1..10..2}
do
echo $i
done
--------------------------------------------------------------------------------
----------
SELECT
------
#! /bin/bash
select değişken in liste
do
  komut
done

#! /bin/bash
select isim in Mehmet Ahmet Veli Ayşe
do
  case $isim in
```

```
  Mehmet )
        echo "Mehmet seçildi";;
  Ahmet )
        echo "Ahmet seçildi";;
  Veli )
        echo "Veli seçildi";;
  Ayşe )
        echo "Ayşe seçildi";;
  * )
        echo "1 ile 4 arasında değer giriniz";;
  esac

done
```

---

Break, Continue ve Until

---

```
#! /bin/bash

for ((i=0;i<=10;i++))
do
    if[ $i -gt 5 ]
    then
    break
    fi
    echo "$i"
done
```

---

```
#! /bin/bash

for ((i=0;i<=10;i++))
do
    if [ $i -eq  2 -o $i -eq 6 ]
    then
    continue
    fi
    echo "$i"
done
```

---

```
#! /bin/bash

i=1

until (($i >= 10))
do
  echo $i
  ((i++)) # i=$((i+1))
done
```

---

FONKSİYONLAR

```
#! /bin/bash

function Merhaba(){

    echo "Merhaba Dostlar"

}

Merhaba
```

---

```
#! /bin/bash

function Merhaba(){

    echo "Merhaba Dostlar"

}

cikis(){

    exit

}


Merhaba

cikis

echo "test"
```

---

```
#! /bin/bash

function Merhaba(){

    echo "Merhaba Dostlar"

}

cikis(){

    exit

}


Merhaba
echo "test"
cikis
```

---

```
#! /bin/bash

echo -e "Bir sayi giriniz:\c "
```

```
read sayi

function Karesiyap(){

    echo "Sayının karesi: $((sayi*sayi))"

}

Karesiyap
------------------------
#! /bin/bash

function cikti(){
    echo $1
}

cikti Ahmet
------------------------
#! /bin/bash

function cikti(){
    echo $1 $2 $3
}

cikti Ahmet evde değil
---------------------------
#! /bin/bash

function cikti(){
    isim=$1
    echo "İsmim $isim"
}

isim="Mehmet"

echo "İsmim $isim"

cikti Ahmet

echo "İsmim $isim"


---------------------------
#! /bin/bash

function cikti(){
    local isim=$1
    echo "İsmim $isim"
}

isim="Mehmet"

echo "İsmim $isim"
```

```
cikti Ahmet

echo "İsmim $isim"

-----------------------------------------------------------------
SHELL'DE KULLANILAN TIRNAKLAR

TERS TIRNAK = `komut`

DÜZ TEK TIRNAK= 'ifade'

DÜZ ÇİFT TIRNAK= "ifade"

-----------------------------------------------------------------
YÖNLENDİRME

3 iletişim kanalı

1 INPUT ----> Klavye < (0)
2 OUTPUT----> EKRAN (Standart output - çıktı) > (1)
3 OUTPUT----> EKRAN (Standart error  - hata)  2> (2)

komut > dosya (Komutun çıktısı dosyaya yazılır)

komut >> dosya (Komutun çıktısı dosya sonuna yazılır)

komut < dosya (Komutun girdisi dosyadan okunur)

komut >| dosya (noclobber set edilmiş olsa dahi komut çıktısı dosyaya
yazılır)

komut 2> dosya (Komutun hataları dosyaya yazılır)

komut > dosya 2>&1 (Komutun çıktısı ve hataları aynı dosyaya yazılır)

komut &> dosya (Komutun çıktısı ve hataları aynı dosyaya yazılır)

komut &>> dosya (Komutun çıktısı ve hataları aynı dosyanın sonuna
eklenir)

komut > dosya1 2> dosya2 (Komutun çıktısı dosya1'e hataları dosya2'ye
yazılır)bir
-----------------------------------------------------------------------
-------

KURALLI İFADELER (REGULAR EXPRESSIONS)

 ^   Satır başı anlamına gelir.
 $   Satır sonu anlamına gelir.
 .   Herhangi bir karakter demektir.
 *   Kendisinden önceki karakterleri tekrarlatır.
 []  Köşeli parantez içerisindeki karakterlerden biri gelebilir.
 [^] Köşeli parantez içerisindeki karakter haricinde bir karakter
gelebilir.
```

```
-------------------------------------------------------------------------------
-------
BASH SCRIPT DEBUGGING
---------------------
#! /bin/bash -x

set -x

sayi=0

set +x

while ((sayi<=10))
do
    sleep 4
    echo $sayi

    ((sayi++))
done
```