

Bash változó ellenőrzések

1. Bash változó ellenőrzése arra, hogy van-e értéke:

-n \$változó név	van értéke	# Set
-z \$változó név	nincs értéke(üres)	# UnSet

Példa :

amikor van értéke a változónak igaz a feltétel

```
if [[ -n $email ]] ; then
    echo a keresett E-mail: $email
else
    echo nincs ilyen E-mail
fi
```

amikor üres a változó értéke lesz igaz a feltétel

```
if [[ -z $email ]] ; then
    echo a keresett E-mail nem létezik
else
    echo a keresett E-mail: $email
fi
```

2. Bash változó ellenőrzése arra, hogy szám-e:

[[\$var = ~ ^ - ? [0-9] + \$]]

A ^ jelzi a bemeneti minta elejét

A - a "-" karakter

A ? jelentése "0 vagy 1 az előző (-)"

A + azt jelenti, hogy "az előzőek közül legalább egy vagy több ([0-9])"

A \$ jelzi a beviteli minta végét

A \. jelzi hogy „.” (pont) karakter

Pédák:

```
#!/bin/bash
echo "adj meg egy számot: "
read val

if [[ "$val" =~ ^[+-]?[0-9]+$ ]]; then
    echo "Ez egy egész szám."
    # mint 1 2 44
elif [[ "$val" =~ ^[+-]?[0-9]+\.$ ]]; then
    echo "Ez egy szöveg."
    # mint 1.2.44
elif [[ "$val" =~ ^[+-]?[0-9]+\.[0-9]*$ ]]; then
    echo "ez egy tizedestört."
    # mint 1.244
else
    echo "Ez valami szöveg."
fi
```

3. Dátum feldolgozása reguláris kifejezéssel. (Közelítő megoldás)

Példafájl: date.txt

Az elvárt kimenet a date.txt alapján:

2018.04.12.

2018/04/12

1999-01-01

1456.06.07.

Egy lehetséges megoldás:

```
egrep --color '^([0-9]{4}[\./-]([0][1-9])|([1][012]))[\./-]([0][1-9])|([12][0-9])|([3][01]))([\./-]?)$' date.txt
```

Magyarázat:

Sor elején kezdődjön

| 4 darab szám

| | Az alábbi karakterek egyike: '.' '/' '-'

| | | Egy jegyű és két jegyű számok 1-től 12-ig

| | | | | 1-9 10-29 30-31 vagy van vagy nincs elválasztójel a végén

| | | | | | | | | Sor végéig tartson az illesztés

| | | | | | | | | |

- - - - - - - - - - - -
| | | | | | | | | | | | | |

```
egrep --color ' ^ [0-9]{4}[\./-] ((([0][1-9])|([1][012]))[\./-] ((([0][1-9])|([12][0-9])|([3][01]))([\./-])?) $' date.txt
```

4. Email cím feldolgozása reguláris kifejezéssel. (Közelítő megoldás)

Példafájl: email.txt

Az elvárt kimenet az email.txt alapján:

jmcmamara@verizon.net.net

emmanuel@yahoo.com

arandal@mac.com

netsfr@outlook.com

moinefou@sbcglobal.net

jmorris@mac.com

kewley@yahoo.info

multiplx@yahoo.com

paina@outlook.com

user@inf.u-szeged.hu

Egy lehetséges megoldás:

```
egrep --color '^([0-9a-z\.-]+)@([0-9a-z]+\.[a-z]{2,4})$' email.txt
```

Magyarázat:

Lehet bármilyen szám, kis betű, majd a végén egy pont

| Ez az előző minta ismétlődhet, minimum egynek lennie kell

| | Hasonló, mint az előző viszont a pont csak a végén lehet, és ez ismétlődhet,

| | úgy hogy min 1 darab legyen, és egy darabon belül min 1 karakter

| | | A végén csak betűk lehetnek

| | | | és azokból is 2 és 4 közötti mennyiségű

| | | | |

| | | | |

```
egrep --color '^([0-9a-z\.-]+) @ ([0-9a-z]+\.[a-z]{2,4})$' email.txt
```

5. Fájlvizsgálat

Az alábbi kapcsolókkal a rendszerben megtalálható fájlokra kérdezhetünk rá.

-b fájlnev	blokkeszköz-meghajtó	# Block	
-c fájlnev	karaktereszköz-meghajtó	# Character	
-d fájlnev	könyvtár	# Directory	
-f fájlnev	szabályos állomány	# File	
-l fájlnev	közvetett hivatkozás	# Link	
-p fájlnev	csővezeték	# Pipe line	
-e fájlnev	létezik	# Exists	😊
-G fájlnev	saját csoportba tartozik	# Group	
-O fájlnev	saját tulajdon	# Own	
-r fájlnev	olvasható	# Read	
-w fájlnev	írható	# Write	
-x fájlnev	futtatható	# eXecute	
-s fájlnev	fájlnev üres	# String	
-h fájlnev	igaz, ha a fájlnev létező szimbolikus link neve		
fájl1 -nt fájl2	a fájl1 újabb, mint a fájl2	# Newer	
fájl1 -ot fájl2	a fájl1 régebbi, mint a fájl2	# Older	
fájl1 -ef fájl2	a fájl1 és fájl2 azonos állományt jelöl	# Equal	