

#### Shell script:

- Olyan szöveges állomány, ami UNIX parancsokat tartalmaz.
- Futtatásához szükséges:
  - Olvasási jog: ez elengedhetetlen.
  - Végrehajtási jog: opcionális.
  - A fájl első sora: `#!/bin/bash` (bash helyén tetszőleges parancsértelmező állhat)
  - A fájl első sorából tudja a burok, hogy script-ről van szó, amelyet a megadott parancsértelmező hajt végre.
  - Ha nincs megadva más, az alapértelmezett parancsértelmező hajtja végre.
- Script futtatása:
  - `bash fájlnev`
  - `bash < fájlnev`
  - `./fájlnev` (végrehajtási jog szükséges)

#### Változók:

- Környezeti változók:
  - SHELL: az alapértelmezett parancsértelmező.
  - PWD: aktuális jegyzék elérési útvonala.
  - PATH: jegyzéklista a parancsok eléréséhez.
  - USER: felhasználónév.
  - stb.
- Pozicionális paraméterek:
  - `$0`, `$1`, `$2`, ..., `$9`: a programnak paraméterként adott paraméterek, `$0` a program neve.
  - `$#`: parancssori paraméterek száma.
  - `$*`, `$@`: a parancssori paraméterek egyben.
  - `$$`: a futó program folyamatazonosítója (PID-je).
  - `$?`: a legutóbb végrehajtott parancs visszatérési értéke.
- Saját változók:
  - Értékadás: `x=2016` (a változónév, az egyenlőségjel, és az érték közé nem teszünk szóközt!).
  - Hivatkozás a saját változóra: `echo $x` (`x` értékét az `stdout`-ra írja).
  - A létrehozott változó ahhoz a shell-hez tartozik, amelyben létrehozzuk (és a gyerekeihez), ha új shell-ből akarjuk elérni a változót, exportálni kell: `export x`.
  - Változó értékének beolvasása az `stdin`-ről: `read x`.

#### Paraméter behelyettesítés:

- Idézőjelek közötti különbségek:
  - `echo 'cat $1'` végrehajtásakor ez íródik ki: `cat $1`
  - `echo „cat $1”` végrehajtásakor ez íródik ki: `cat paraméter_értéke`



- o `echo `cat $1`` végrehajtásakor a paraméterként kapott fájl tartalma íródik ki. (ez az eredmény így is elérhető: `echo $(cat $1)`)
- `${változó:-érték}` ha a változó értéke nem üres string, akkor az fejtődik ki, egyébként pedig az érték. (default paraméter beállítására használható)
- `${változó:=érték}` ha a változó értéke üres string, felveszi az értéket.
- `${változó:?érték}` ha a változó értéke nem üres string, kifejtődik, egyébként kiíródik az érték és exitál a shell.
- `${változó:+érték}` ha a változó értéke nem üres string, behelyettesítődik az érték.

#### Egész számok kezelése:

- Vegyünk két változót, mindkettőnek egy-egy számértéket adunk, pl.: `x=10` és `y=2`.
- Összeadás eredményének kiírása:
  - o `echo „$(x + y)”`
  - o `expr $x + $y`
- Kivonás eredményének kiírása:
  - o `echo „$(x - y)”`
  - o `expr $x - $y`
- Szorzás eredményének kiírása:
  - o `echo „$(x * y)”`
  - o `expr $x \* $y`
- Osztás eredményének kiírása:
  - o `echo „$(x / y)”`
  - o `expr $x / $y`
- Valós számok kezeléséhez szükségünk van egy segédprogramra, például a „bc”-re. Használatára példa: `echo „5.27 + 4.73” | bc`

#### Stringek kezelése:

- A változóknak adhatunk string értékeket is, pl.: `x="Hello World!"`.
- String hosszának kiírása:
  - o `echo „${#x}”`
  - o `expr length „$x”`
- Substring kiírása:
  - o `echo „${x:6}”`
  - o `echo „${x:6:5}”`
  - o `expr substr „$x” 6 5`
- Karakter string-ben való első előfordulásának indexe:
  - o `expr index „$x” W`
  - o Egy karakter helyett karakterlánc is megadható, ez esetben a karakterlánc valamely karaktere első előfordulásának indexét adja.
- Karakterlánc lecserélése másik karakterláncra:
  - o `echo ${x/"World"/"Alex"}`

- Ha a lecserélni kívánt karakterlánc többször is előfordul a stringben és mindegyiket le akarjuk cserélni: `echo ${x//\"World\"/\"Alex\"}`

Tömbök kezelése:

- Létrehozhatunk tömböket a következőképpen: `x=(érték1 érték2 ...)`
- Az értékeket a zárójelen belül szóközzel választjuk el egymástól.
- A tömbök nullás indexelésűek, esetünk x 0. eleme érték1.
- Hivatkozás a tömb elemeire:
  - Írásnál: `x[2]=érték3`
  - Olvasásnál: `echo ${x[2]}`
- Tömb bővítése elemekkel: `x=(${x[*]} érték3 érték4 ...)`
- Tömb összes elemének kiírása: `echo ${x[*]}` vagy `echo ${x[@]}`
- Tömb elemszámának kiírása: `echo ${#x[*]}` vagy `echo ${#x[@]}`
- Tömb adott elemeinek kicserélése: `echo ${x[*]}/érték2/érték4}`
- Tömb létrehozása másképp: `declare -a x=(érték1 érték2 ...)` (az inicializálás elmaradhat)
- Tömb törlése: `unset x`