

05

BASH script programozás I.

# BASH (man bash)

- ez egy shell (Bourne Again SHell): hozzáférést biztosít a kernel funkcióihoz, kényelmi szolgáltatások, programok indítása:  
**parancsértelmező**
- chsh : felhasználó shelljéne változtatása (de mi ezt használjuk, ez az alapértelmezett shell)

# BASH script

- script kötegelte végrehajtása:
  - a parancsértelmező nem a parancssorból, hanem a fileből olvassa soronként a parancsokat
- saját parancsot hozhatunk létre (automatizálás)
- felépítése:
  - `#!/bin/bash` első sor, parancsértelmező fejrész
  - lehetőség van bármilyen parancs (akár program) végrehajtására
  - soronként egy parancs
    - de lehet többsoros parancs is sorvégi: `\`
    - lehet egy sorban több „sornyi” parancs is: `;` (sortörés szimbólum)
  - változók, vezérlési szerkezetek (if, for, while, ...)
  - **!!!nagyon kötött a szintaxis!!!**

# BASH script alapok

- Parancsértelmező fejrész: `#!/bin/bash`
- Egysoros kommentelés: `#`
- 'SZÖVEG':
  - nincs jelentése a spec karaktereknek
  - pl nem lehet változóra hivatkozni, nincs parancsbehelyettesítés
- "SZÖVEG": jelentése van a spec karaktereknek:
  - `$` változónév hivatkozás
  - `` `` közé írt parancs végrehajtódik és az eredmény behelyettesítődik
  - `\` kezdve a fenti spec karakter is kiírható, pl: `\$`

Feladat: nézzük meg a `behelyettesites.sh` -t

# Szöveg kiírása

- echo – kiír egy stringet a képernyőre (kapcsolóval escape szekvenciák)
- printf "FORMAT" "PARAMÉTEREK" – C-hez hasonló módon escape szekvenciákkal (pl: \n, \t, %s, stb ..) értelmezi és kiírja a szöveget  
Példa: printf "Hello %s\n" "World"

# Változók

- értékadás: VALTOZO\_NEV=érték  
(!!nincs szóköz az = után !!!)
  - hivatkozás:  
    \$VALTOZO\_NEV  
    vagy \${VALTOZO\_NEV}
  - beolvasás standard bemenetről:  
    read VALTOZO\_NEV
  - névadás: betűket, számokat és \_ jelet tartalmazhat, de nem kezdődhet számmal, hagyomány szerint nagybetűsek
  - érték: **szöveg**
- Feladat:** Kérjünk be egy szöveget, adjuk át egy változóba majd írassuk ki

# Parancsmegadás

- **szokásos** módon parancs megadása (mint ahogy a parancssorban megszoktuk)
  - ekkor végrehajtódik és a STDOUT/ERR-ra íródik a kimenet
- **hivatkozás** a parancs eredményére:
  - ekkor a parancs lefut és az eredménye fog behelyettesítődni és megjelenni az adott sorban
  - `PARANCS`
  - **$\$(PARANCS)$**

Feladat: Is kimenetét adjuk át egy változónak és írassuk ki

# Matematikai jelölés

- **(( KIF ))**

- csak egész számokon értelmezett, eredmény is egész (használhatóak változók)
- +, -, \*, /, %, ++, --, \*\*<sub>(hatványozás)</sub>, !, ~, &, ^, &&, ||, ?:, =, +=, -=, \*=, /=
- < <= > >= == != (igaz: 1, hamis: 0)
- lehet hivatkozásként is használni: \$(( KIF ))



# Paraméterek

- hivatkozás: \$1, \$2, .... , \$9, \${10}, \${11}, ....
- összes paraméter listában: \$\*
- a script neve: \$0
- paraméterek száma: \$#

Feladat: írjunk egy olyan scriptet, ami

- kiírja az első sorba az összes paramétert
- a következő sorban pedig összeadja az második és harmadik paramétert

# Feladat

- Írj szkriptet, amely két parancssori paramétert vár. Mindkét paraméter egy-egy szövegfájl legyen, melyek első sora egy-egy számot tartalmaz. A szkript írja ki ennek a két számnak az összegét a képernyőre.