

*Skriptiranje v*

*#!/bin/bash*

*Spremenljivke*

# ***Spremenljivke***

- Deklaracija ni potrebna.
- Tip spremenljivke?
- Definicija.
  - *ime=vrednost*
  - **local** *ime=vrednost*
  - **readonly** *ime=vrednost*
  - **export** *ime=vrednost*

# ***Spremenljivke***

- Uporaba spremenljivke v skripti
  - $\$ \{ime\}$  – osnovni način;
  - $\$ime$  – krajši način.
- Privzete vrednosti
  - $\$ \{ime : -privzeto\}$  – vsebina ali privzeta vrednost;
  - $\$ \{ime := privzeto\}$  – nastavitev vrednosti.

# ***Spremenljivke***

- Delo s podnizi
  - `${#ime}` – dolžina *niza*;
  - `${ime:poz}` – podniz od *poz* do konca;
  - `${ime:poz:dol}` – podniz na *poz* dolžine *dol*.
- \*Posredna razširitev
  - `${!ime}` – kadar spremenljivka vsebuje ime spremenljivke;
  - `a=123; b=a; echo $b ${!b}`



# ***Spremenljivke***

- Odstranjevanje podnizov
  - $\${ime\#vzorec}$  – najkrajša predpona;
  - $\${ime\#\#vzorec}$  – najdaljša predpona;
  - $\${ime\%vzorec}$  – najkrajša pripona;
  - $\${ime\%\%vzorec}$  – najdaljša pripona.
- Zamenjava podniza
  - $\${ime/vzorec/vrednost}$  – prva pojavitev
  - $\${ime//vzorec/vrednost}$  – vse pojavitve

# ***Spremenljivke***

- \*Zanimive spremenljivke:
  - `$SHELL`, `$BASH`, `$BASH_VERSION`,
  - `$SHELLOPTS`,
  - `$LINES`, `$COLUMNS`, `$LINENO`,
  - `$PWD`, `$OLDPWD`,
  - `$RANDOM`, `$SECONDS`.

# ***Uporaba navednic***

- Enojne in dvojne navednice.
  - Če niz vsebuje presledke.
  - Enojne vzamejo niz dobessedno.
  - Dvojne izvedejo morebitne substitucije.

```
a=Triglav  
echo '$a'  
echo "$a"
```

```
a='Martin          Krpan'  
echo $a  
echo "$a"
```



*Skriptiranje v*

*#!/bin/bash*

*Pogojni izrazi*



# ***Pogojni izrazi***

- Ukaz **test** *pogoj*.
- Enojni oglati oklepaji: [ *pogoj* ].
  - Za oba ukaza obstaja vgrajena in zunanja različica.
  - Oba ukaza prejmeta argumente (pazljivo s presledki),  
le da ukaz [ kot zadnji argument zahteva ].
  - Priporočena uporaba dvojnih navednic okoli spremenljivk.
- Izhodni status: \$?

# ***Pogojni izrazi***

- Dvojni oglati oklepaji: `[ [ pogoj ] ]`.
  - Ni ukaz, ampak ključna beseda, zato je sintaktično manj občutljiv.
  - Omogoča več kot **test**, npr. ujemanje vzorcev.
  - Obstaja v lupini **bash**, ne pa v **sh**.

# ***Pogojni izrazi***

- Primerjava nizov.
  - **test -z niz** – je dolžina *niza* enaka nič;
  - **test -n niz** – je dolžina *niza* večja od nič;
  - **test niz1 == niz2** – sta *niza* enaka;
  - **test niz1 != niz2** – sta *niza* različna;
  - **test niz1 < niz2** – je *niz1* pred *niz2*;
  - **test niz1 > niz2** – je *niz1* za *niz2*.



# ***Pogojni izrazi***

- Preverjanje datotek.
  - **test -e** *datoteka* – ali datoteka obstaja;
  - **test -f** *datoteka* – je navadna datoteka;
  - **test -d** *datoteka* – je imenik;
  - **test -L** *datoteka* – je simbolična povezava;
  - **test -b** *datoteka* – je bločno orientirana;
  - **test -c** *datoteka* – je znakovno orientirana;
  - itd.



# Sestavljanje ukazov

- Pogojno izvajanje
  - Izhodni status;
  - *ukaz1* && *ukaz2* – način „in“;
  - *ukaz1* || *ukaz2* – način „ali“
  - Kratkostična evaluacija izraza;
  - Ukaza **true** in **false**.

TRUE

FALSE

```
true || echo "Se tole izpiše?"  
false && echo "Se tole izpiše?"  
cd ~ && mkdir test && cd test && touch juhuhu.txt || echo "Težave na vidiku."
```

# Sestavljanje ukazov

- Združevanje ukazov
  - { *ukazi*; }
- Pozor:
  - obvezen presledek za predklepajem;
  - obvezen presledek pred zaklepajem; in
  - obvezno podpičje za zadnjim ukazom.

```
mkdir test && { cd test; touch dat.txt; echo Zivjo; }  
test -f /etc/passwd && { echo Imam jo.; cat /etc/passwd; } || echo Napaka
```

# *Izvajanje v podlupini*

- Podlupina je podproces trenutne lupine.
- Navadno:
  - *( ukaz )*
- S substitucijo:
  - *\$ ( ukaz )*
- Spremenljivka **\$BASH\_SUBSHELL**.

```
( echo Zivjo; exit 42; echo svet! )  
dirname $(which less)  
( echo $BASH_SUBSHELL ); echo $BASH_SUBSHELL
```



# Aritmetika

- Dvojni oklepaji:
  - `(( aritmetika ))`, `$( ( aritmetika ) )`
  - Sintaksa podobna jeziku C, ni težav s presledki.
  - Pred spremenljivkami ne potrebujemo `$`.

```
(( a = 1 + 2 * 3 ))  
a=$(( 1 + 2 * 3 ))  
(( b=a**a, a++, c = a + a, a += 2 ))  
(( a <= 12 )) && echo Malo || echo Veliko  
echo $a $b $c $(( 42 * 24 ))
```