LOCAL VARIABLES & TYPESET

Thursday, 6 May 2021 4:57 PM

USING AWK: https://www.geeksforgeeks.org/awk-command-unixlinux-examples/

Local Variables and Typeset

- Variables can be created in a function that will not be available outside of it.
- The typeset command makes variables local, can provide a type, or can provide formatting.

```
 typeset -i x
# x must be an integer
```

Local Variables and Typeset

- Arithmetic is faster for variables defined to be integers.
- Let allows for convenient arithmetic:
  `let x++; let y=x**2; let x=x*3; let x*=5, ...`

DECLARE COMMAND:

The Declare Command

- `declare -l` uppercase values in the variable are converted to lowercase.
- `declare -u` lowercase values in the variable are converted to uppercase.
- `declare -r` variable is made read-only.

The Declare Command

- `declare -a MyArray` will make *MyArray* an indexed array.
- `declare -A MyArray2` will make *MyArray2* an associative array.

READ COMMAND:

The Read Command

The Read Command

- Read a line into a variable or multiple variables
- read a b—reads first word into a and the rest into b
- Convenient for a while loop

WHILE LOOPS:

While Loops

```
while
  ((x<10))
do
  echo loop $x; date >data.$x
 ((x=x+1))
done
```

READ IS COMING OUT OF DATA FILE:

While Loops

```
while
  read a b
do
  echo a is $a b is $b
```

```
  echo a is $a b is $b
done <data_file
```

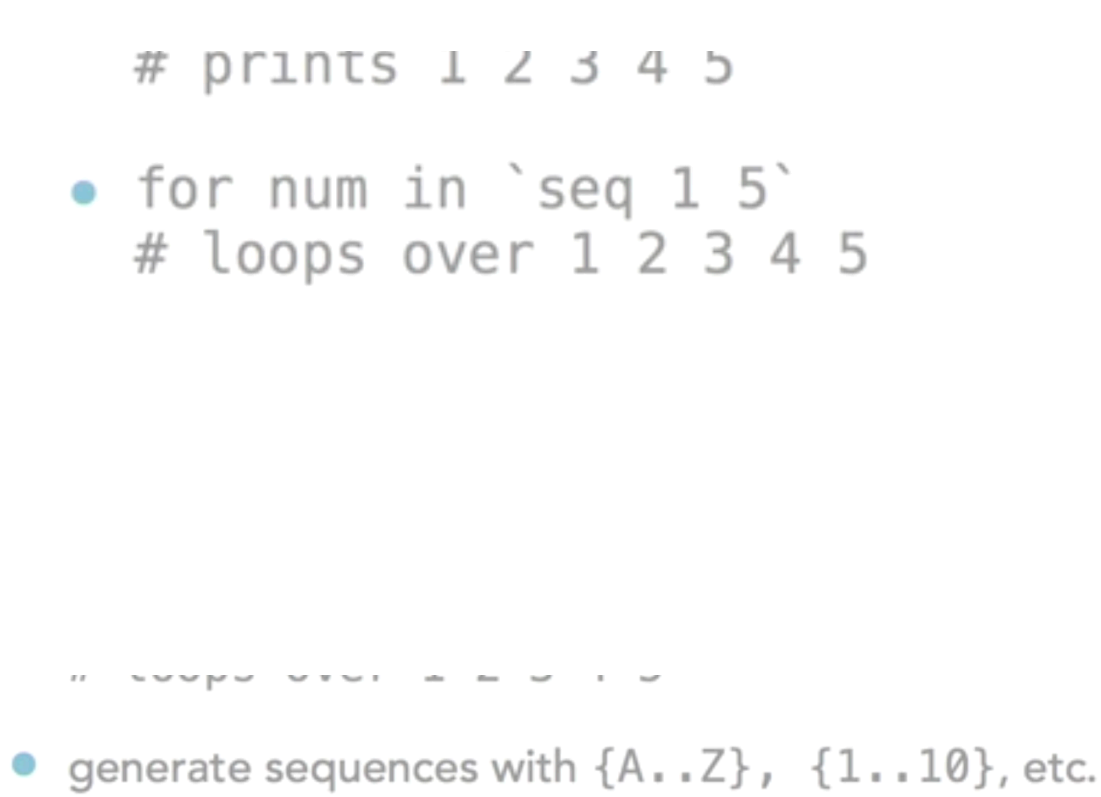
While Loops

```
ls -l | while
    read a b c d
    do
      echo owner is $c
    done
```

SEQ COMMAND IN FOR LOOPS:

For Loops

- seq 1 5



OTHER WAY TO USE FOR LOOPS:

We can use `` back ticks or $ sign to pass a command as an argument to the loop `