

```
#####
#no spaces when using the assignment operator in shell
## # No of arguments passed at command line
$0 # name of script
$1 $2 $3 # variables passed at runtime
whoami
date
ln #create hard link
ln -s #create symbolic link
# use varname when assigning value to variables, use $varname when accessing variable data
who #who is logged in
echo ${#varAry[*]} or echo ${#varAry[#]} #echos number of values in array
echo ${#varAry[i]} #echos length of element at index i
#5 last page for user and group checking. also check user creating script
#6 for info on conditional operators
#8 maths notes, performing calculations within loops
```

```
#####
while [ $i -le 4 ]; do #note spaces inside []
    # do something here
done
```

```
#####
if [ $age -gt 0 -a $age -le 1 ] #if and again note spaces in [ ]
then
    #
else
    #
fi
```

```
#####
until [ $count -gt 5 ]; do
    echo "value of count is $count"
    let count=count+1 #note no spaces between assignments
    echo "Please hit enter"
    read
done
```

```
#####
#expressions
echo -n Input a number:
read celsius
fah=expr $celsius * 1.8 + 32
echo Temperature in fahrenheit is: -n
$fah
```

```
#####
planets=(Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune Pluto)
```

```
for p in ${planets[*]}
do
    echo This planet is $p
done
```

```
#####
while true
do
    echo -n "Enter a positive number: "
    read myNumber

    if [ $myNumber -gt 0 ]
    then
        break
    else
        echo "Not a positive number, please try again"
    fi
fi
```

done

```
echo "You have entered number $myNumber"
```

```
#####
```

```
options=("Check if a user exists" "Create user" "Delete user" "Check if a group exists" "Create group"
"Delete group" "Exit")
```

```
while true
```

```
do
```

```
    select option in "${options[@]}" # careful with this syntax, make sure { and } are in the right
    places
```

```
    do
```

```
        case $option in
```

```
            "Check if a user exists")
```

```
                #do something
```

```
                break;
```

```
        ;;
```

```
            "Create user")
```

```
                #do something
```

```
                break;
```

```
        ;;
```

```
            "Delete user")
```

```
                #do something
```

```
                break;
```

```
        ;;
```

```
            "Check if a group exists")
```

```
                #do something
```

```
                break;
```

```
        ;;
```

```
            "Create group")
```

```
                #do something
```

```
                break;
```

```
        ;;
```

```
            "Delete group")
```

```
                #do something
```

```
                break;
```

```
        ;;
```

```
            "Exit")
```

```
                exit
```

```
        ;;
```

```
        *)
```

```
            echo "Invalid option selected, try again"
```

```
        ;;
```

```
    esac
```

```
done
```

```
done
```

```
#####
```

```
# list-glob.sh: Generating [list] in a for-loop, using "globbing"
```

```
for file in *          # The wild card character "*" matches every filename,
```

```
do
```

```
    ls -l "$file" # Lists all files in current director
```

```
done
```

```
echo; echo
```

```
for file in [a]*
```

```
do
```

```
    rm -f $file # Removes only files beginning with "a" in the current dir
```

```
    echo "Removed file $file".
```

```
done
```

```
echo
```

exit 0

#####