

Practice 4

Sample 1

The following sample is similar to last weeks' practice, but gives some examples of variables again.

```
#!/bin/bash
#note that when assigning variables you should never have spaces on
# either side of the =
name="Jay and Silent Bob"
echo $name

echo "How old are you?"; read age

echo "You input $age"

a=`ls -l /var/log/apache2`
echo $a

u=`cat /etc/passwd | grep joe | awk -F\: '{ print $1}'`
echo $u
```

A few things to pay attention to:

- Is there a difference between `$u` and `${u}`? Try the following exercise:

```
a="hello world"
echo $a
echo ${a}
echo $a_something
echo ${a}_something
```

- What did you discover?

You should now be confident in setting variables, passing in command-line arguments and storing them as variables, and asking the user for information and saving it in a variable. If not, keep practicing.

As a general rule, you should always use the `{}` syntax when referring to a variable (like `echo ${myvar}`)

Sample 2

The following will check to see if a machine is running. If so, it will ssh to the machine and change the password for the user `joe`. A simple if statement:

```
newpwd='Foobar123?'
oldpwd='Hello123??'
m="144.38.220.18"
#is that machine up and running?
ping -c1 -W1 $m > /dev/null
#what does this if statement do?
#checks to see if the results of the previous command was successful
if [ $? -eq 0 ]
then
  #indentation doesn't matter but enhances readability
  echo "$m Tis up"
  # change the password for the user
  # we can't run the passwd command because that is interactive
  subcmd="echo \"joe:$newpwd\" | chpasswd"
  #putting your sudo password in a command like this works, but is terrible form
  #can you think of a better way?
  cmd="echo '$oldpwd' | sudo -Sk sh -c '$subcmd'"
  ssh $m -o StrictHostKeyChecking=no $cmd < /dev/null
else
  echo "$m Tis not up"
fi
```

Things to note:

- `echo $?`, try it out. It will show the exit status of a previous command. For example, try to cat out a file that doesn't exist, then use this command to look at the exit status. Try again with a file that does exist.

Sample 3

Write a bash script that prompts the user for their age. If they are over 60, print out "Ok Boomer!". Otherwise if they are over 35, print out "Millennial". Otherwise print out "Young blood."

After you have written your solution or if you get stuck, you can check mine [here](#)