Bash Script Demos

Benjamin Brewster

readloop

```
$ cat readloop
#!/bin/bash
while read myLine
do
        echo "$myLine"
done < $1
$ readloop test_file
1       1       1       1       1
9       3       4       5       5
6       7       8       9       7
3       6       8       9       1
3       4       2       1       4
6       4       4       7       7
$
```

This line stuffs the contents of the file, whose name is the string in the first argument, into the while loop!

Purpose: Read each line in a file, then echo it back out

arrayloop

- Bash arrays have complicated syntax!

```
$ cat arrayloop
#!/bin/bash
array=( one two three )
for i in "${array[@]}"
do
        echo $i
done
$ arrayloop
one
two
three
$
```

Declares and fills array

array[n] fetches n-th element in array
array[@] fetches ALL elements in array

Purpose: Output the contents of an array

forloop

```
$ cat forloop
#!/bin/bash
oneline="1    2    3    4    5"
for i in $oneline
do
        echo "i is: $i"
done
$ forloop
i is: 1
i is: 2
i is: 3
i is: 4
i is: 5
```

Tab delimited, but works for spaces, too

Could also sum them up, etc. here

Purpose: Manipulate each number on a single line, when the length of that line is unknown

sumloop

```
$ cat sumloop
#!/bin/bash
sum=0
TMP1=./tempfile
echo -e "8\n7\n6" > $TMP1

while read num
do
        echo "In Loop"
        echo "num: $num"
        sum=`expr $sum + $num`
        echo "sum: $sum"
        echo -e "End of Loop\n"
done < $TMP1
```

One number per line

Unlike readloop, this is not $1; this uses the filename specified above for input

```
$ sumloop
In Loop
num: 8
sum: 8
End of Loop

In Loop
num: 7
sum: 15
End of Loop

In Loop
num: 6
sum: 21
End of Loop
```

Purpose: Sum up the numbers in a file consisting of one number per line

stdinread

```
$ cat stdinread
#!/bin/bash
cat > "tempfile"
cat tempfile
$ stdinread
I like cheese
I like cheese
$ cat test_file | stdinread
1       1       1       1       1
9       3       4       5       5
6       7       8       9       7
3       6       8       9       1
3       4       2       1       4
6       4       4       7       7
```

The terminal waits here for input, only continuing when you hit CTRL-D (EOF)

Purpose: Capture data from stdin and direct it to a temp file

trtest

```
#!/bin/bash
# This script converts a row file ./tempinputfile into a column file ./tempcolfile,
# then back into a row file ./temprowfile<PID>
inputFile="tempinputfile"
tempCol="tempcolfile"
tempRow="temprowfile"

# Make the input row file
echo -e "1\t2\t3\t4\t5" > $inputFile

# Append each number onto the end of a temporary column file by cutting specific columns
cut -c 1 $inputFile > $tempCol
cut -c 3 $inputFile >> $tempCol
cut -c 5 $inputFile >> $tempCol
cut -c 7 $inputFile >> $tempCol
cut -c 9 $inputFile >> $tempCol

# Convert the column file back into a row file
cat $tempCol | tr '\n' '\t' > "$tempRow$$"

# Add a newline char to the end of the row file, for easier printing
echo >> "$tempRow$$"
```

Purpose: Convert a row file into a column file and back again