Bash Scripting Practice and Q&A

Alper Kinaci
Sr. Computational Specialist
NUIT, Research Computing

Northwestern

Bash Scripting Practice

- What is a Bash script?

Bash commands in a file

- Why should I write Bash scripts?

To create command workflows and reuse them frequently

Bash Scripting Practice

We will solve problems to practice

- Variables and arguments
- Loops
- Conditionals
- Arrays
- Functions

Google and command help documents are your best friend when learning command line and Bash scripting.

Bash Scripting Practice

Prerequisites

Data:

https://swcarpentry.github.io/shell-novice/data/data-shell.zip

Bash Shell:

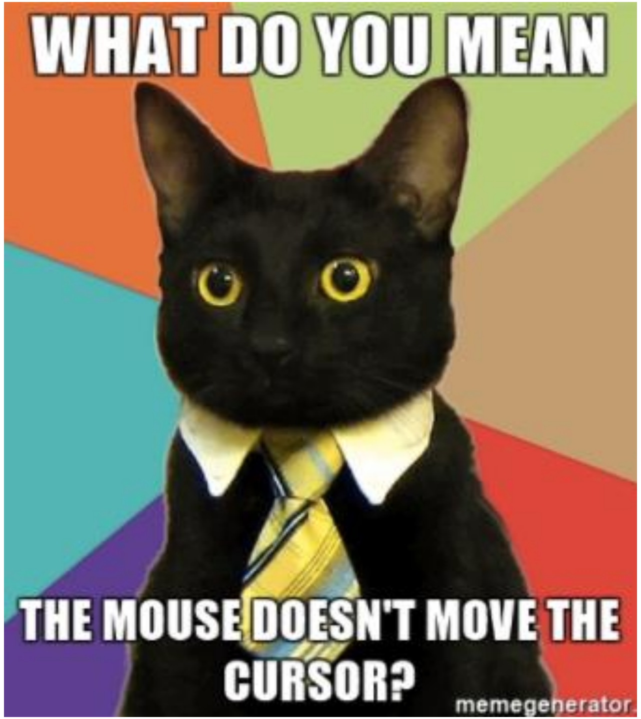
Windows – Git Bash https://gitforwindows.org/

Mac – issue "chsh -s /bin/bash", close and reopen the terminal

Bash Scripting Practice

Text Editor: Nano

- $ nano <scriptname>
  Opens or creates the file
- type your text: the mouse will not move the cursor - navigate with the arrow keys
- save and quit: commands are on the bottom of the screen - the “^” stands for “control”



Bash Scripting Practice

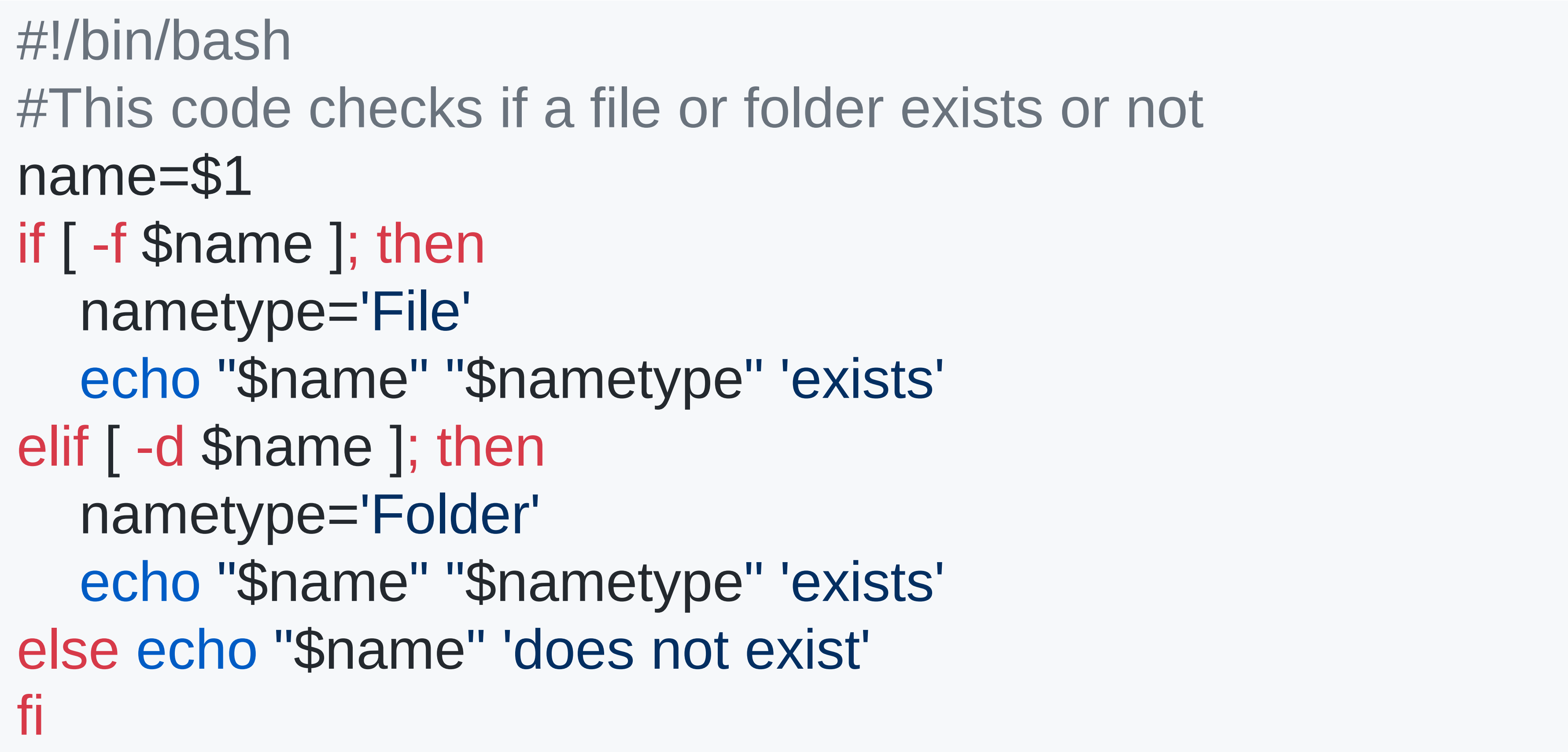
Exercises and answers can be downloaded from:

https://github.com/nuitrcs/bash-scripting-practice

Exercises

1- *Concepts: Variables, arguments, conditionals*

Write a script to find if a file or folder exists in "./data-shell/" directory. Input the name as a parameter and if the file/folder exits, the script should print out "<name> file/folder exists". Otherwise the script should print out "<name> does not exits"

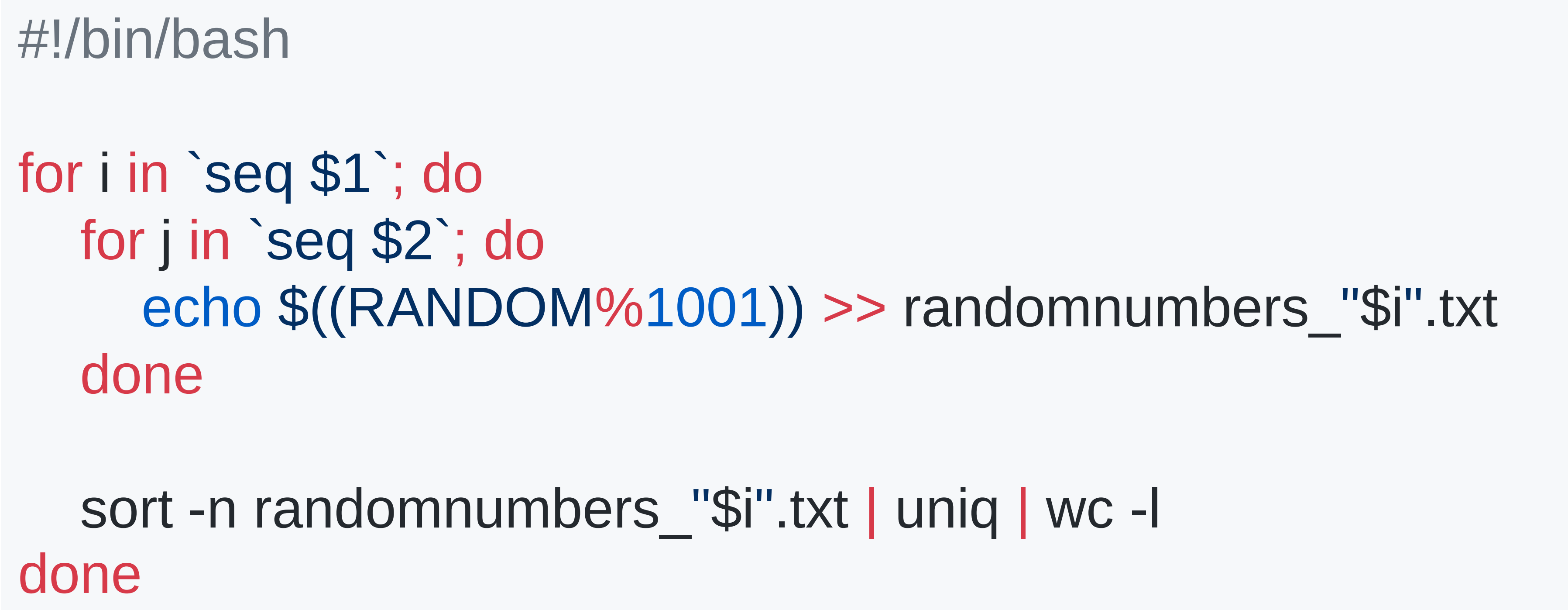


```
#!/bin/bash
#This code checks if a file or folder exists or not
name=$1
if [ -f $name ]; then
    nametype='File'
    echo "$name" "$nametype" 'exists'
elif [ -d $name ]; then
    nametype='Folder'
    echo "$name" "$nametype" 'exists'
else echo "$name" 'does not exist'
fi
```

Exercises

2- *Concepts: Variables, arguments, nested loops*

Write a bash script to generate N random numbers between 0 and 1000 and write them to a distinct file. Find out how many unique numbers are obtained. Repeat all this M times. N and M are input parameters for the script.

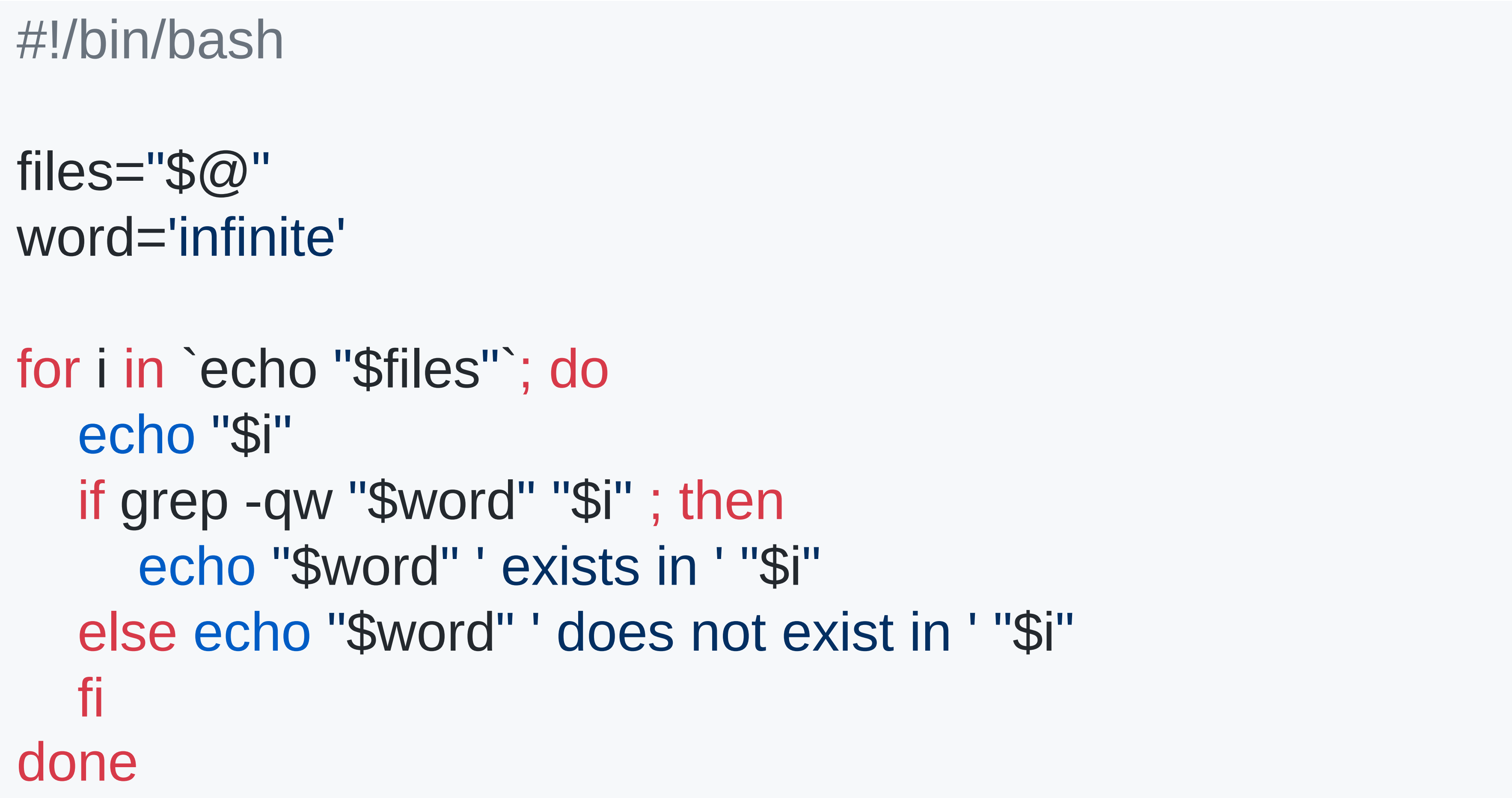


```bash
#!/bin/bash

for i in `seq $1`; do
    for j in `seq $2`; do
        echo $((RANDOM%1001)) >> randomnumbers_"$i".txt
    done

    sort -n randomnumbers_"$i".txt | uniq | wc -l
done
```

Exercises

3- *Concepts: Variables, arguments, nested loops, conditionals*

Write a bash script to search all the files in "/data-shell/writing/data" folder and let us know if the 'infinite' is in the file or not for each file.

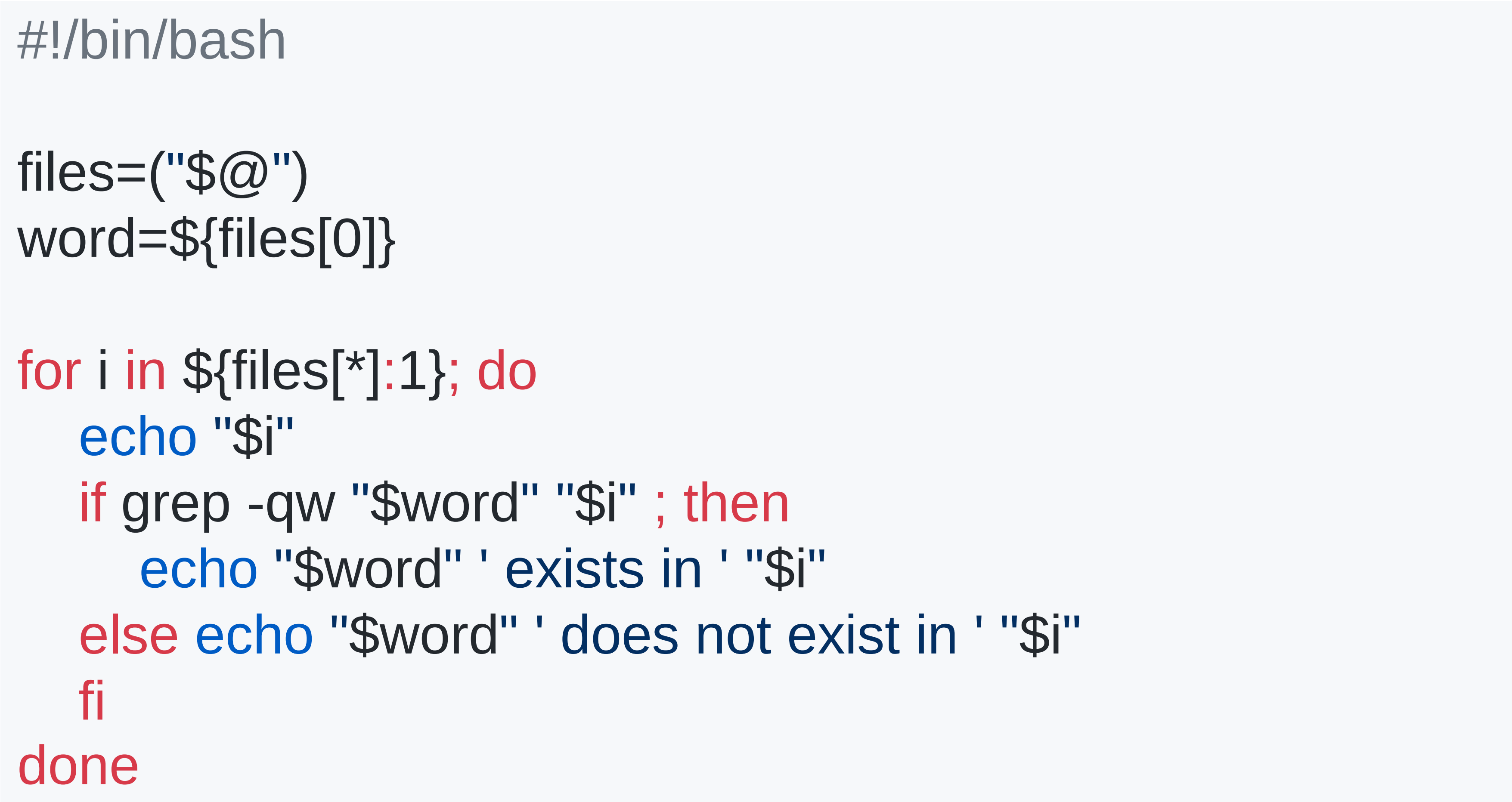


```
#!/bin/bash

files="$@"
word='infinite'

for i in `echo "$files"`; do
    echo "$i"
    if grep -qw "$word" "$i" ; then
        echo "$word" ' exists in ' "$i"
    else echo "$word" ' does not exist in ' "$i"
    fi
done
```

Exercises

4- *Concepts: Variables, arguments, nested loops, conditionals, arrays*

Write a bash script to search all the files in "/data-shell/writing/data" folder and let us know if an input word is in the file or not for each file.

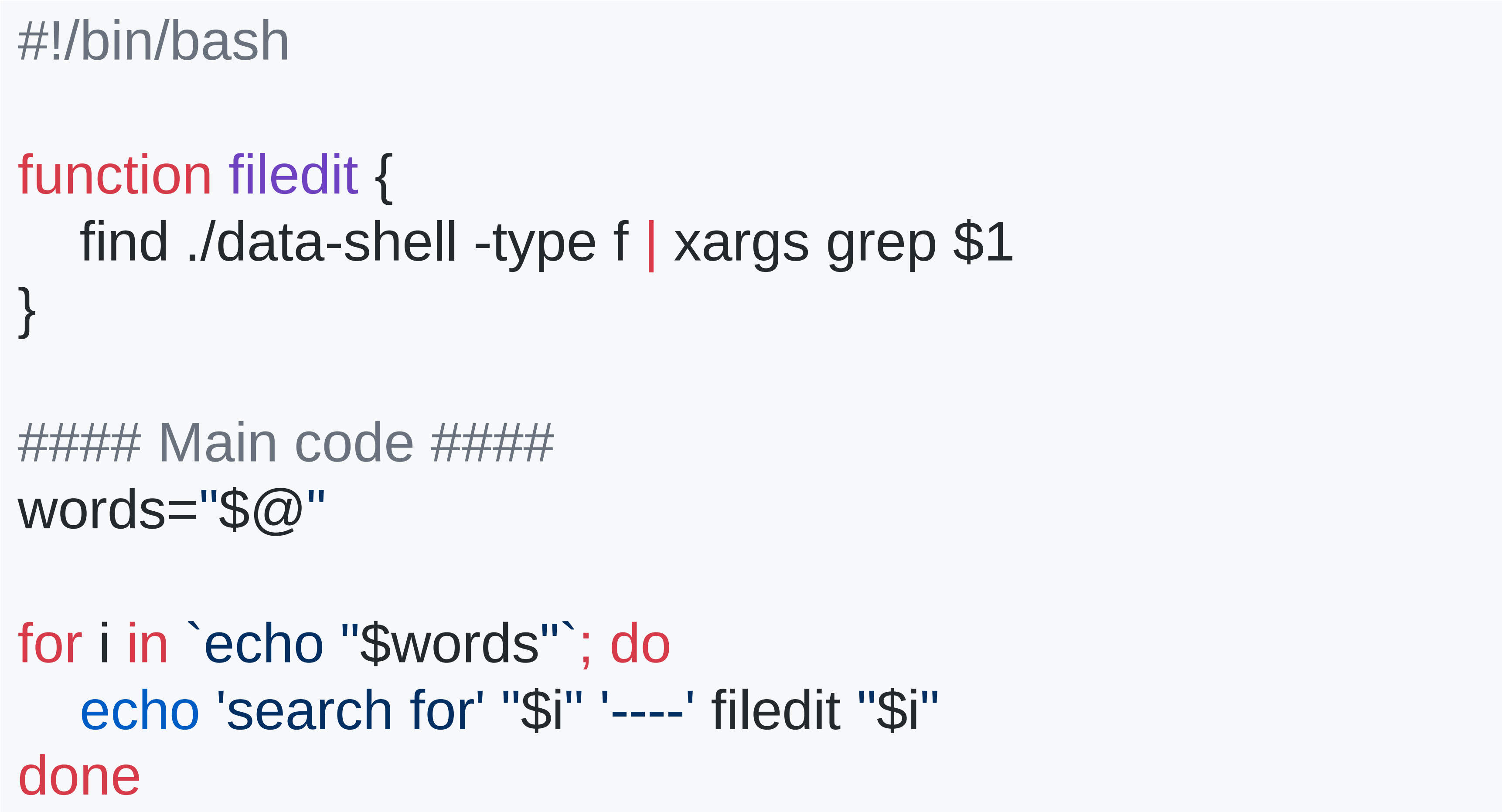


```
#!/bin/bash

files=("$@")
word=${files[0]}

for i in ${files[*]:1}; do
    echo "$i"
    if grep -qw "$word" "$i" ; then
        echo "$word" ' exists in ' "$i"
    else echo "$word" ' does not exist in ' "$i"
    fi
done
```

Exercises

5- *Concepts: Variables, arguments, loops, functions*

Write a function to search for input strings in files in the current directory and all the directories within.



```bash
#!/bin/bash

function filedit {
    find ./data-shell -type f | xargs grep $1
}

##### Main code #####
words="$@"

for i in `echo "$words"`; do
    echo 'search for' "$i" '----' filedit "$i"
done
```

Exercises

6- *Concepts: Variables, arguments, loops, conditionals, arrays, functions*

Write a bash which takes two arguments. The first argument can be 'folder' or 'file'.

- For 'folder', the second argument should be the path of a folder
- For 'file', the second argument should be the path of a file.

If folder is entered, the script should operate on all files in that folder. If a file is entered, the script should operate only on that file. The operation include replacing a line that has a matching string and deleting the 4th line in the file(s).

```
#!/bin/bash function
filedit {
  echo "processing $1"
  # substitute a whole line with a matching string
  sed -e 's/.*DAVE.*/REPLACE/' $1 > $1_version1
  # delete 4th line in the file
  sed -e '4d' $1 > $1_version2
  # add more contents, if required
  echo "EDITED on 07/22" >> $1_version1
  echo "EDITED on 07/22" >> $1_version2
}
```

```
#### Main code ####
args=("$@")
currentfolder=$PWD
if [ "${args[0]}" == ‘folder' ]; then
  if [ -d "${args[1]}" ]; then
    cd ${args[1]}
    for i in `ls`; do
      filedit $i
    done
  else echo "${args[1]}" 'is not a directory'
  fi
  cd $currentfolder
elif [ "${args[0]}" == 'file' ]; then
  if [ -f "${args[1]}" ]; then
    filedit ${args[1]}
  else echo "${args[1]}" 'is not a file'
  fi
else echo 'Please enter all or file for the first argument'
fi
```

Q&A