BASH

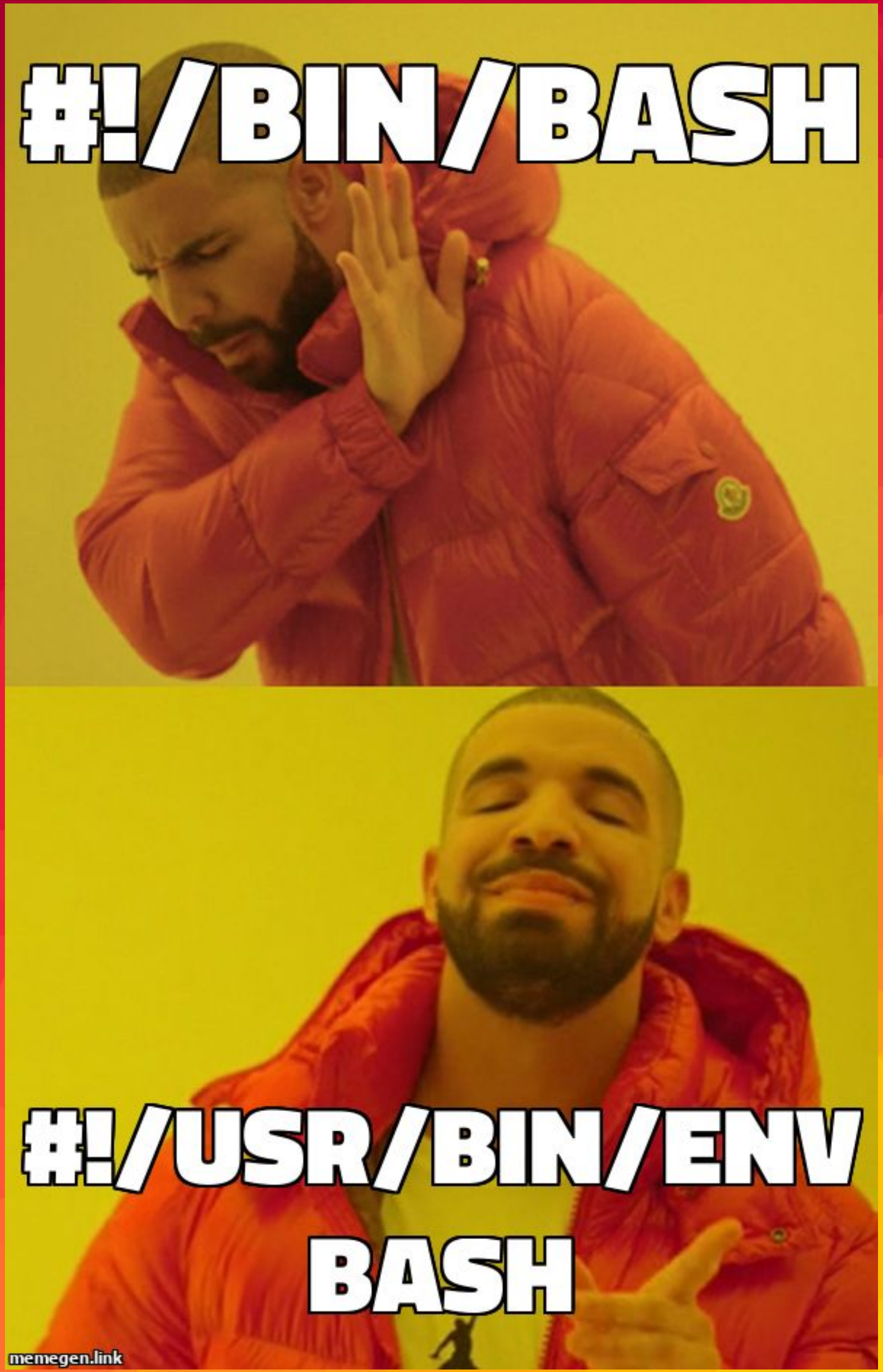
Seizing The Memes Of Production

BASH

- Unix Shell & Command Language
- Bourne Again Shell
- Written by Brian Fox in 1989 to replace the Bourne Shell
- https://www.gnu.org/software/bash/

Meme Generator Script

- Stripped-down version of this script we found
- Original author: Navan Chauhan (github.com/navanchauhan)
- Generates memes using memegen.link/
- Source code: git.redbrick.dcu.ie/tbolt/bash-talk-2018



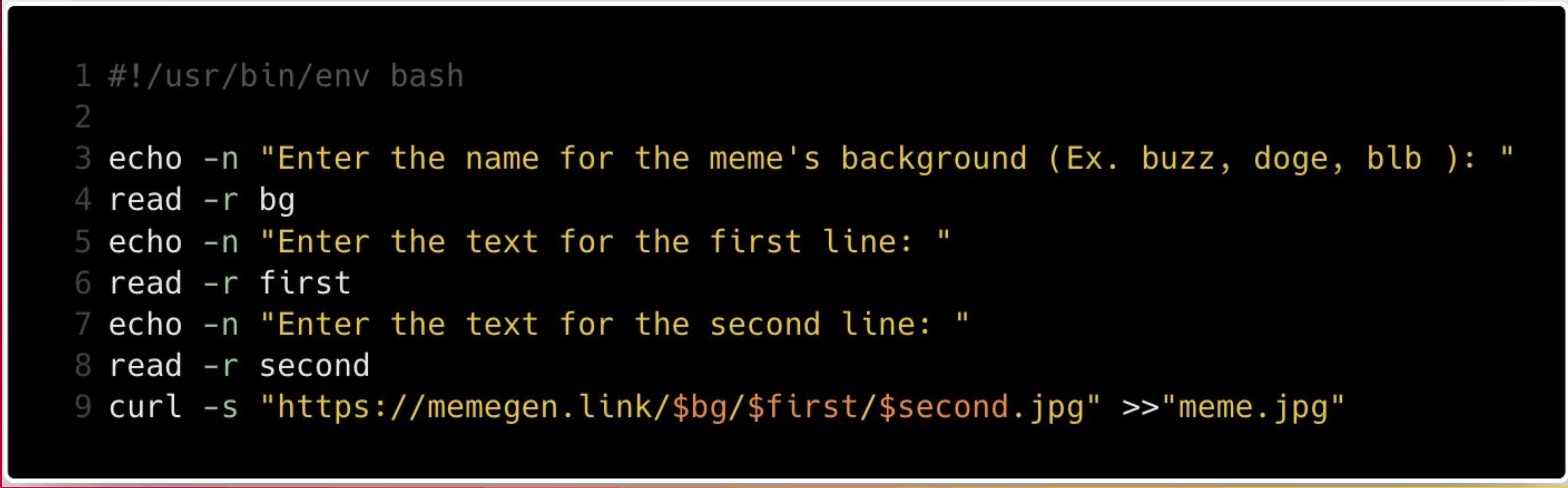
Redbrick
DCU's Networking Society

Some notes

- In some cases, particularly on DCU machines, the .jpg extension doesn't work. Swap for .png in this case.
- If at first your script doesn't work, don't be disheartened, one of us will help you out!

First iteration

- Prompt user for meme information
- Variables
- Fetch meme using curl
- File redirection



```bash
#!/usr/bin/env bash

echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
read -r bg
echo -n "Enter the text for the first line: "
read -r first
echo -n "Enter the text for the second line: "
read -r second
curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"meme.jpg"
```

Redbrick
DCU's Networking Society

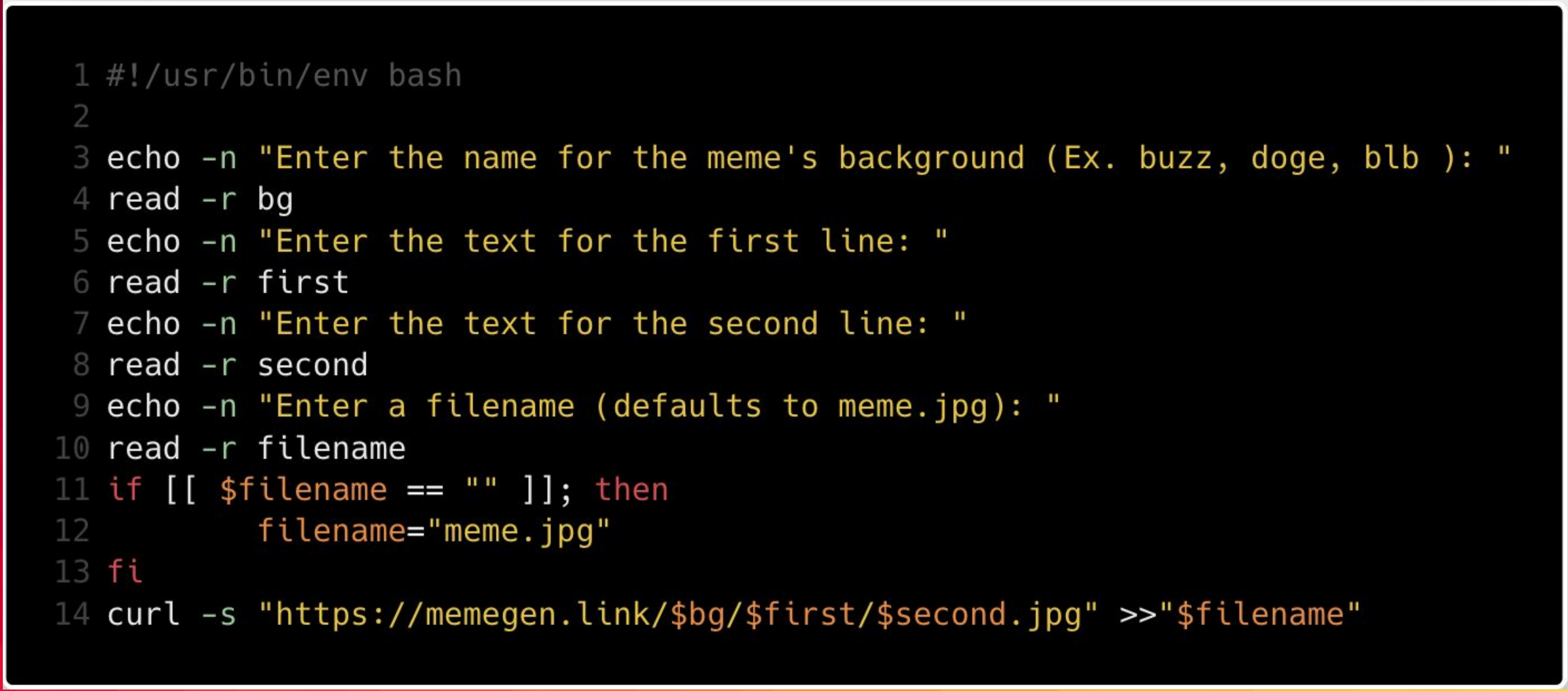
How to run your script

- Save your script as *meme.sh*
- On the command line run *chmod +x meme.sh*
- Run *./meme.sh* on the command line.

Redbrick
DCU's Networking Society

Second iteration

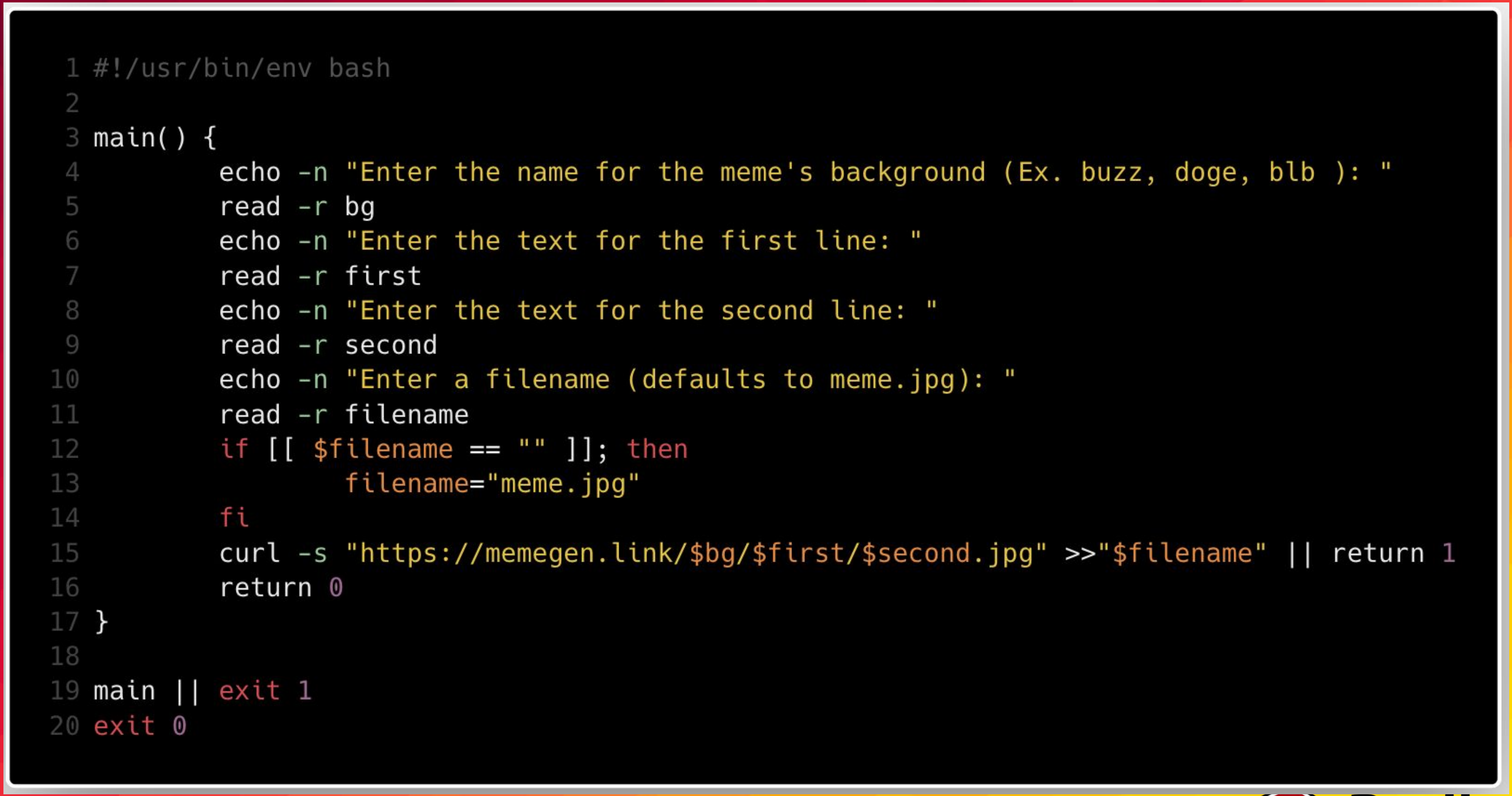
- Let the user change the file name
- Give a default file name if they don't provide one
- Conditionals



```bash
#!/usr/bin/env bash

echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
read -r bg
echo -n "Enter the text for the first line: "
read -r first
echo -n "Enter the text for the second line: "
read -r second
echo -n "Enter a filename (defaults to meme.jpg): "
read -r filename
if [[ $filename == "" ]]; then
        filename="meme.jpg"
fi
curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"$filename"
```

Third iteration

- Make the program "scriptable"
- But first we need a main function

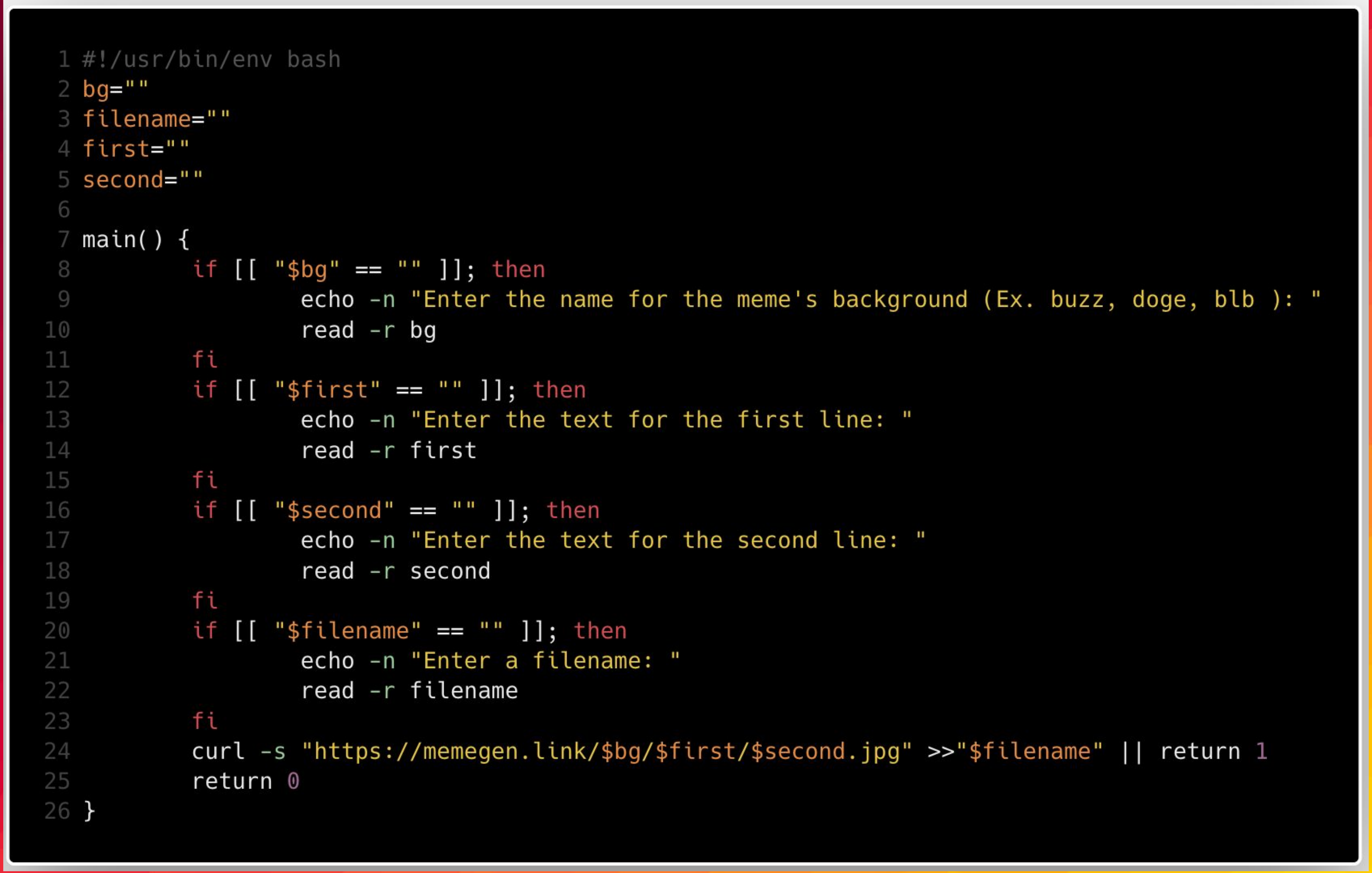


```bash
#!/usr/bin/env bash

main() {
        echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
        read -r bg
        echo -n "Enter the text for the first line: "
        read -r first
        echo -n "Enter the text for the second line: "
        read -r second
        echo -n "Enter a filename (defaults to meme.jpg): "
        read -r filename
        if [[ $filename == "" ]]; then
                filename="meme.jpg"
        fi
        curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"$filename" || return 1
        return 0
}

main || exit 1
exit 0
```

Redbrick
DCU's Networking Society

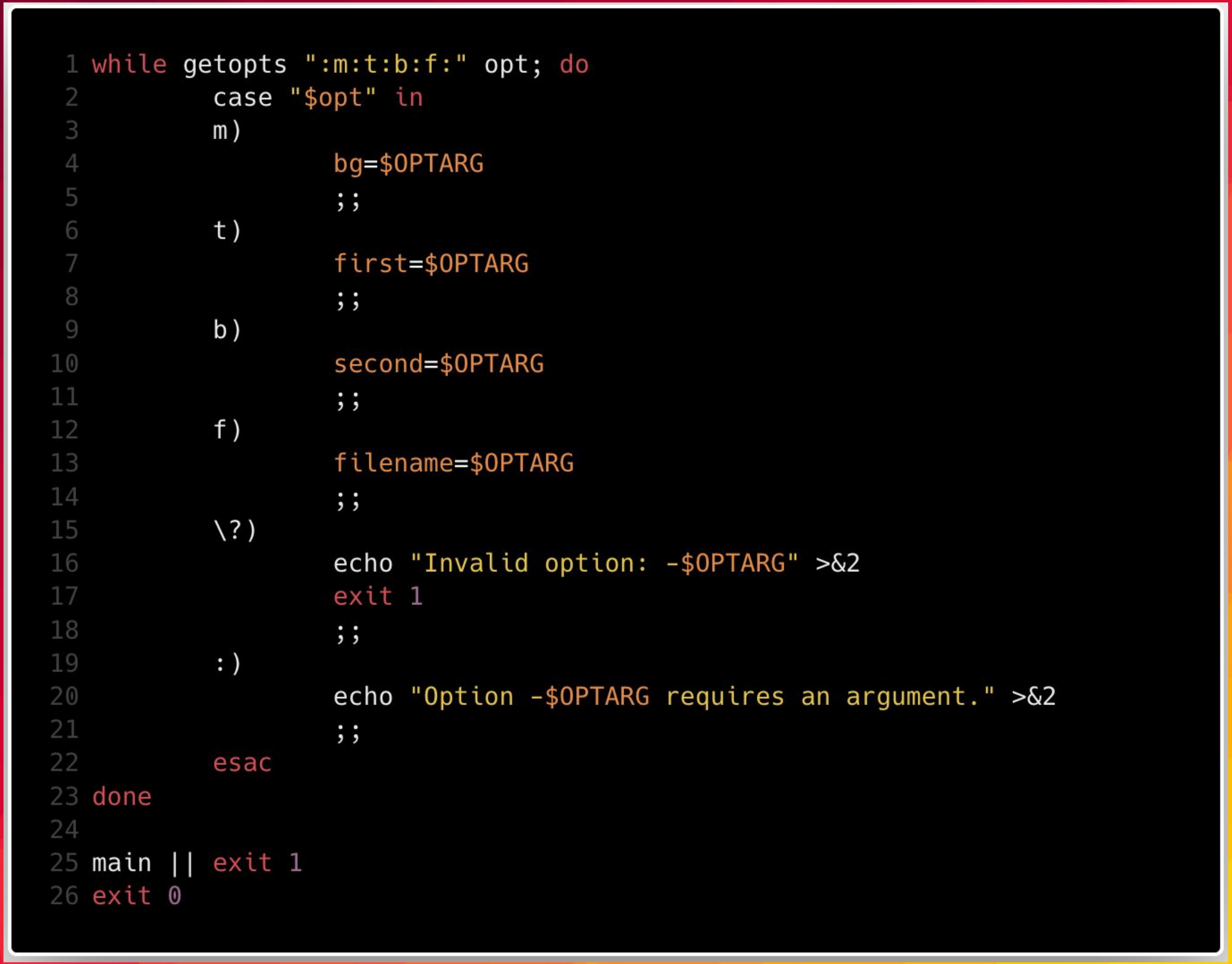
Fourth iteration

- Make it only prompt the user if it has to
- Add arguments using getopts



```bash
#!/usr/bin/env bash
bg=""
filename=""
first=""
second=""

main() {
        if [[ "$bg" == "" ]]; then
                echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
                read -r bg
        fi
        if [[ "$first" == "" ]]; then
                echo -n "Enter the text for the first line: "
                read -r first
        fi
        if [[ "$second" == "" ]]; then
                echo -n "Enter the text for the second line: "
                read -r second
        fi
        if [[ "$filename" == "" ]]; then
                echo -n "Enter a filename: "
                read -r filename
        fi
        curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"$filename" || return 1
        return 0
}
```

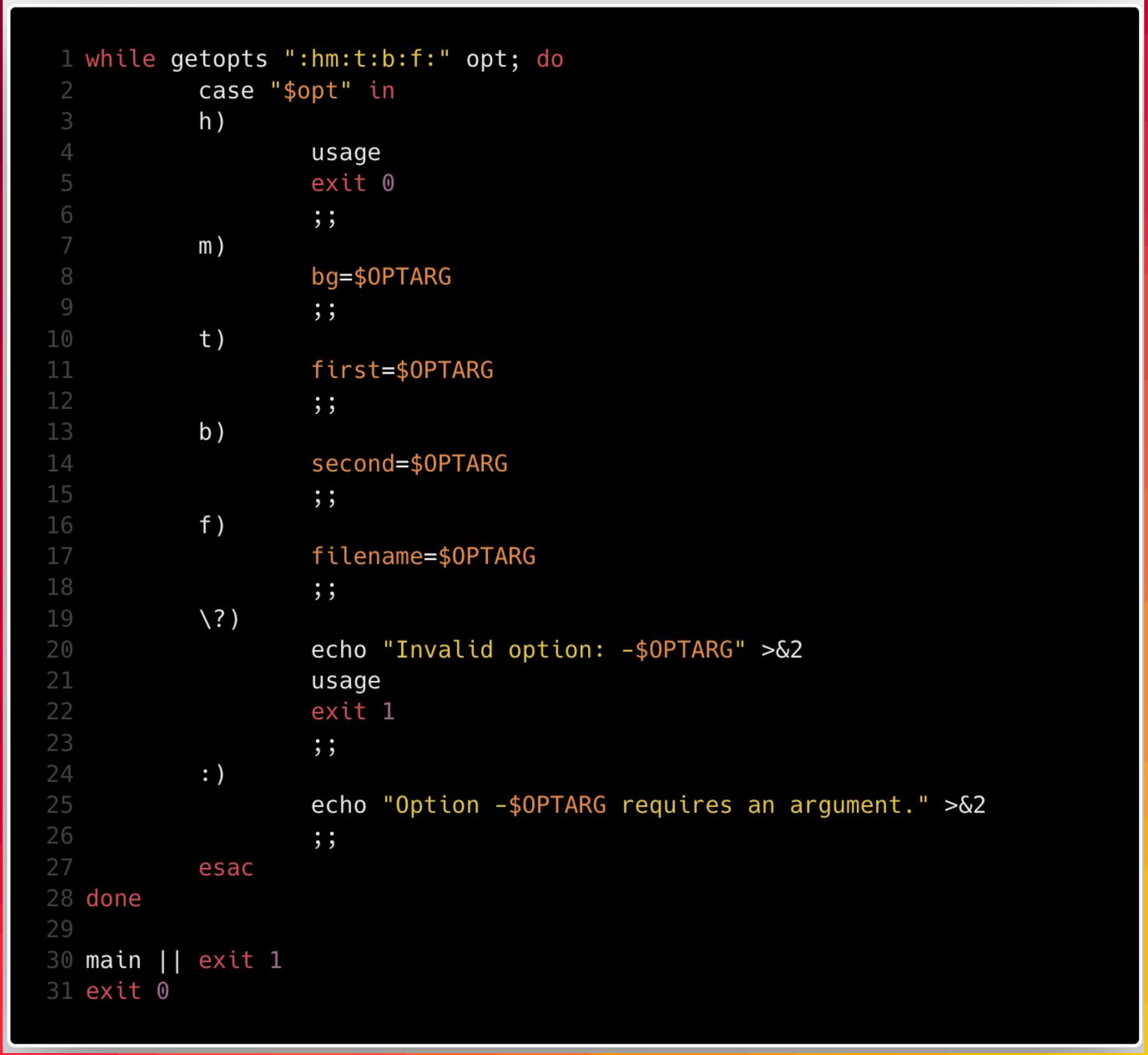
Redbrick
DCU's Networking Society



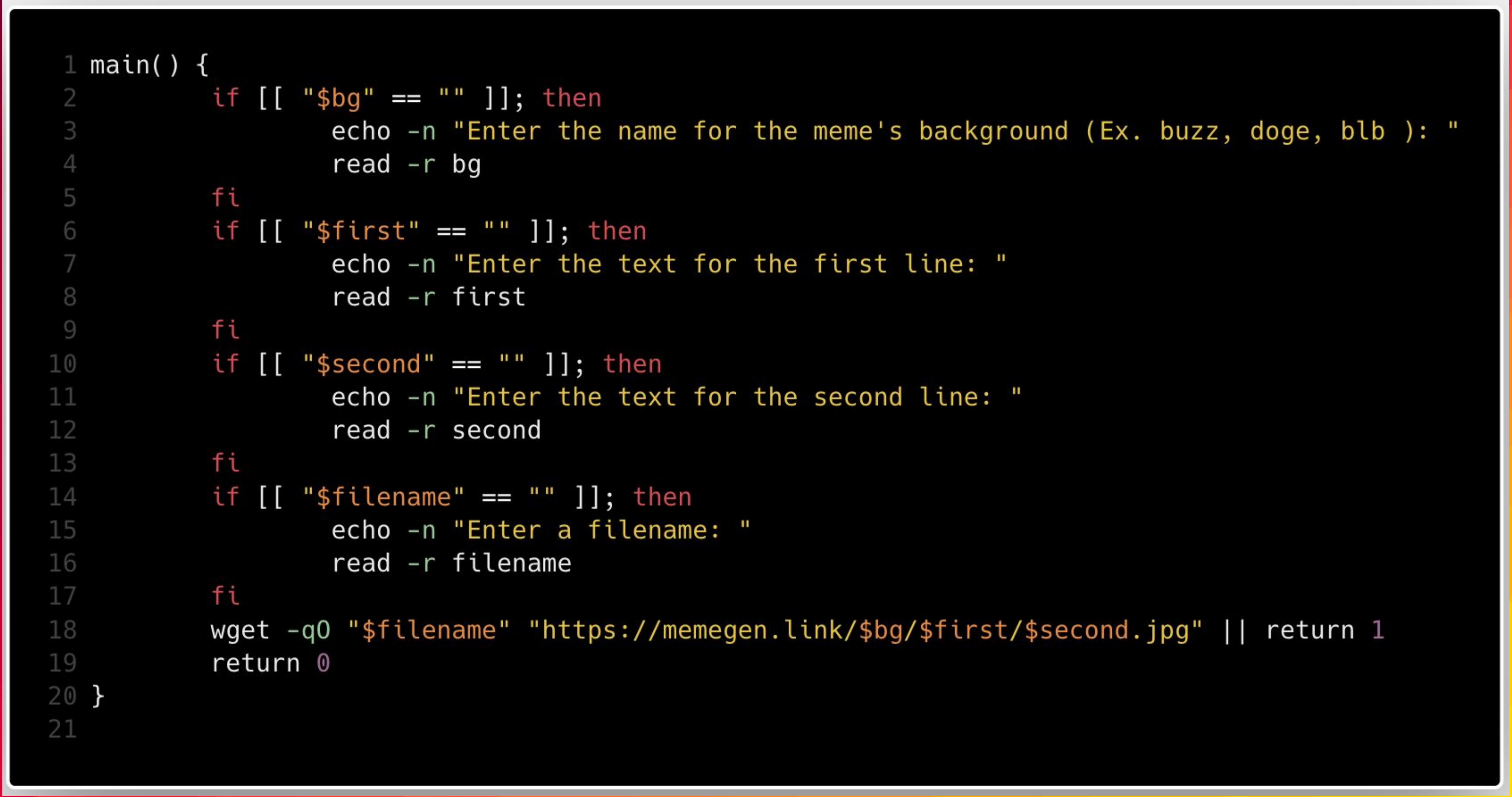
```
while getopts ":m:t:b:f:" opt; do
        case "$opt" in
        m)
                bg=$OPTARG
                ;;
        t)
                first=$OPTARG
                ;;
        b)
                second=$OPTARG
                ;;
        f)
                filename=$OPTARG
                ;;
        \?)
                echo "Invalid option: -$OPTARG" >&2
                exit 1
                ;;
        :)
                echo "Option -$OPTARG requires an argument." >&2
                ;;
        esac
done

main || exit 1
exit 0
```

Final iteration

- Usability
- Usage string
- Cat and EOF
- Swapping out curl for wget

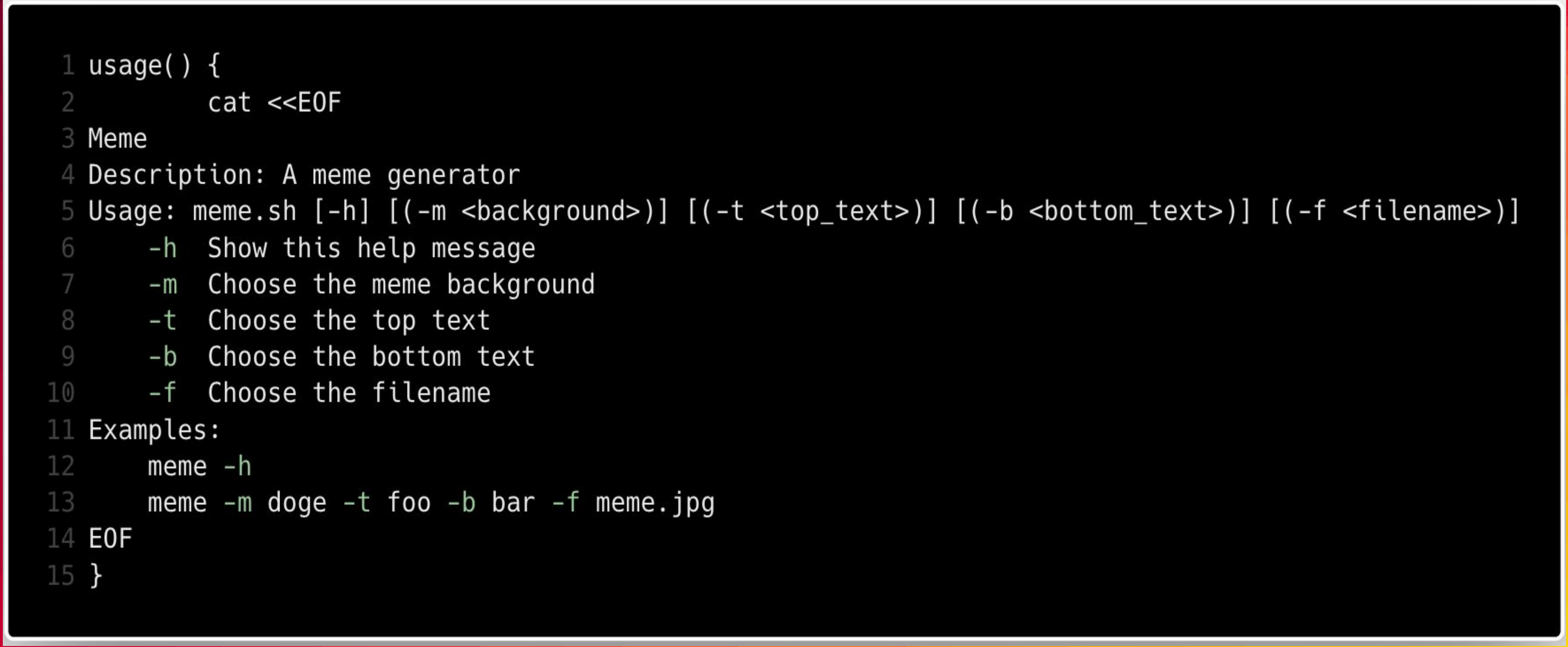


```
while getopts ":hm:t:b:f:" opt; do
        case "$opt" in
        h)
                usage
                exit 0
                ;;
        m)
                bg=$OPTARG
                ;;
        t)
                first=$OPTARG
                ;;
        b)
                second=$OPTARG
                ;;
        f)
                filename=$OPTARG
                ;;
        \?)
                echo "Invalid option: -$OPTARG" >&2
                usage
                exit 1
                ;;
        :)
                echo "Option -$OPTARG requires an argument." >&2
                ;;
        esac
done

main || exit 1
exit 0
```



```
main() {
        if [[ "$bg" == "" ]]; then
                echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
                read -r bg
        fi
        if [[ "$first" == "" ]]; then
                echo -n "Enter the text for the first line: "
                read -r first
        fi
        if [[ "$second" == "" ]]; then
                echo -n "Enter the text for the second line: "
                read -r second
        fi
        if [[ "$filename" == "" ]]; then
                echo -n "Enter a filename: "
                read -r filename
        fi
        wget -qO "$filename" "https://memegen.link/$bg/$first/$second.jpg" || return 1
        return 0
}
```

Redbrick
DCU's Networking Society



```
usage() {
        cat <<EOF
Meme
Description: A meme generator
Usage: meme.sh [-h] [(-m <background>)] [(-t <top_text>)] [(-b <bottom_text>)] [(-f <filename>)]
    -h  Show this help message
    -m  Choose the meme background
    -t  Choose the top text
    -b  Choose the bottom text
    -f  Choose the filename
Examples:
    meme -h
    meme -m doge -t foo -b bar -f meme.jpg
EOF
}
```

Redbrick
DCU's Networking Society

Demo

Redbrick
DCU's Networking Society