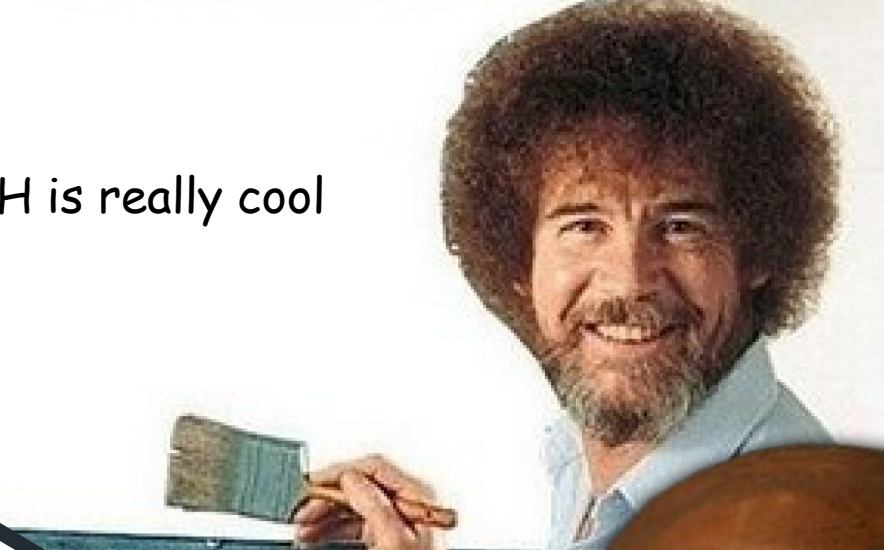




BASH is really cool



# 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](https://github.com/navanchauhan))
- Generates memes using [memegen.link/](https://memegen.link/)
- Source code: [git.redbrick.dcu.ie/tbolt/bash-talk-2018](https://git.redbrick.dcu.ie/tbolt/bash-talk-2018)



**#!/BIN/BASH**



**#!/USR/BIN/ENV  
BASH**



[memegen.link](https://memegen.link)



**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

```
1 #!/usr/bin/env bash
2
3 echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
4 read -r bg
5 echo -n "Enter the text for the first line: "
6 read -r first
7 echo -n "Enter the text for the second line: "
8 read -r second
9 curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"meme.jpg"
```





# 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.





# Second iteration

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

```
1 #!/usr/bin/env bash
2
3 echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
4 read -r bg
5 echo -n "Enter the text for the first line: "
6 read -r first
7 echo -n "Enter the text for the second line: "
8 read -r second
9 echo -n "Enter a filename (defaults to meme.jpg): "
10 read -r filename
11 if [[ $filename == "" ]]; then
12     filename="meme.jpg"
13 fi
14 curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"$filename"
```



# Third iteration

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



```
1 #!/usr/bin/env bash
2
3 main() {
4     echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
5     read -r bg
6     echo -n "Enter the text for the first line: "
7     read -r first
8     echo -n "Enter the text for the second line: "
9     read -r second
10    echo -n "Enter a filename (defaults to meme.jpg): "
11    read -r filename
12    if [[ $filename == "" ]]; then
13        filename="meme.jpg"
14    fi
15    curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"$filename" || return 1
16    return 0
17 }
18
19 main || exit 1
20 exit 0
```



# Fourth iteration

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

```
1 #!/usr/bin/env bash
2 bg=""
3 filename=""
4 first=""
5 second=""
6
7 main() {
8     if [[ "$bg" == "" ]]; then
9         echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
10        read -r bg
11    fi
12    if [[ "$first" == "" ]]; then
13        echo -n "Enter the text for the first line: "
14        read -r first
15    fi
16    if [[ "$second" == "" ]]; then
17        echo -n "Enter the text for the second line: "
18        read -r second
19    fi
20    if [[ "$filename" == "" ]]; then
21        echo -n "Enter a filename: "
22        read -r filename
23    fi
24    curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"$filename" || return 1
25    return 0
26 }
```





```
1 while getopts ":m:t:b:f:" opt; do
2     case "$opt" in
3         m)
4             bg=$OPTARG
5             ;;
6         t)
7             first=$OPTARG
8             ;;
9         b)
10            second=$OPTARG
11            ;;
12         f)
13            filename=$OPTARG
14            ;;
15         \?)
16            echo "Invalid option: -$OPTARG" >&2
17            exit 1
18            ;;
19         :)
20            echo "Option -$OPTARG requires an argument." >&2
21            ;;
22     esac
23 done
24
25 main || exit 1
26 exit 0
```



# Final iteration

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

```
1 while getopts ":hm:t:b:f:" opt; do
2     case "$opt" in
3         h)
4             usage
5             exit 0
6             ;;
7         m)
8             bg=$OPTARG
9             ;;
10        t)
11            first=$OPTARG
12            ;;
13        b)
14            second=$OPTARG
15            ;;
16        f)
17            filename=$OPTARG
18            ;;
19        \?)
20            echo "Invalid option: -$OPTARG" >&2
21            usage
22            exit 1
23            ;;
24        :)
25            echo "Option -$OPTARG requires an argument." >&2
26            ;;
27    esac
28 done
29
30 main || exit 1
31 exit 0
```



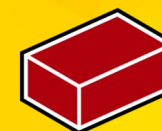


```
1 main() {
2     if [[ "$bg" == "" ]]; then
3         echo -n "Enter the name for the meme's background (Ex. buzz, doge, blb ): "
4         read -r bg
5     fi
6     if [[ "$first" == "" ]]; then
7         echo -n "Enter the text for the first line: "
8         read -r first
9     fi
10    if [[ "$second" == "" ]]; then
11        echo -n "Enter the text for the second line: "
12        read -r second
13    fi
14    if [[ "$filename" == "" ]]; then
15        echo -n "Enter a filename: "
16        read -r filename
17    fi
18    wget -qO "$filename" "https://memegen.link/$bg/$first/$second.jpg" || return 1
19    return 0
20 }
21
```



```
1 usage() {
2     cat <<EOF
3 Meme
4 Description: A meme generator
5 Usage: meme.sh [-h] [(-m <background>)] [(-t <top_text>)] [(-b <bottom_text>)] [(-f <filename>)]
6     -h Show this help message
7     -m Choose the meme background
8     -t Choose the top text
9     -b Choose the bottom text
10    -f Choose the filename
11 Examples:
12    meme -h
13    meme -m doge -t foo -b bar -f meme.jpg
14 EOF
15 }
```

# Demo



**Redbrick**  
DCU's Networking Society