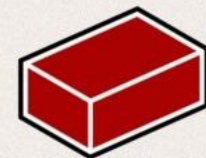


# BASH

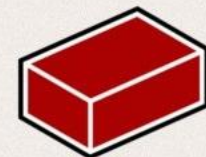
Seizing The Memes Of Production



**Redbrick**  
DCU's Networking Society

# Bash

- Unix Shell & Command Language
- **B**ourne **A**gain **S**hell
- 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 <https://memegen.link/>
- Source code: <https://git.redbrick.dcu.ie/tbolt/bash-talk-2018>



# First iteration

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

```
#!/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: "
```

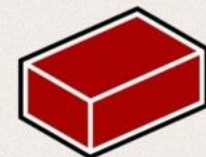
```
read -r first
```

```
echo -n "Enter the text for the second line: "
```

```
read -r second
```

```
curl -s "https://memegen.link/$bg/$first/$second.jpg" >>"meme.jpg" || return 1
```

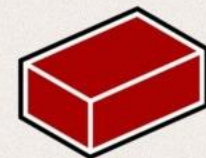
```
return 0
```



**Redbrick**  
DCU's Networking Society

# Second iteration

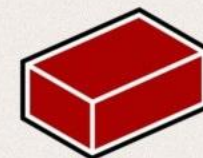
- Let the user change the file name
- Give a default file name if they don't provide one
- Conditionals





```
#!/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" || return 1
return 0
```



# Third iteration

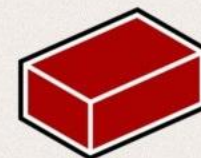
- Make the program "scriptable"
- But first we need a main function



```
#!/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
}

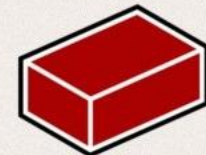
main || exit 1
exit 0
```



# Fourth iteration

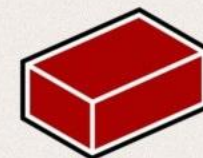
- Make it only prompt the user if it has to
- Add arguments using getopt

```
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  
}
```





```
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
```

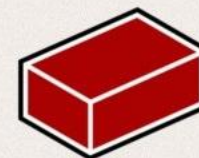


**Redbrick**  
DCU's Networking Society

# Final iteration

- Usability
- Usage string
- Cat and EOF

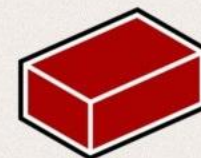
```
while getopts ":hm:t:b:f:" opt; do
    case "$opt" in
        h)
            usage
            exit 0
            ;;
        m)
            bg=$OPTARG
            ;;
        t)
            first=$OPTARG
            ;;
        b)
            second=$OPTARG
            ;;
        f)
            filename=$OPTARG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



**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  
}
```



# Demo