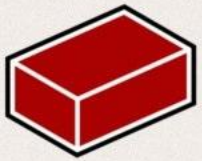
Intro to Bash Scripting

#!/bin/bash

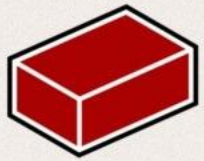


Redbrick

DCU's Networking Society

Scripts for todays talk at:

https://github.com/redbrick/HelpdeskTalks



Redbrick

DCU's Networking Society

Shell

- Bash is a shell
- **B**ourne **A**gain **Sh**ell
- Bash can read commands from a file
  - and this is scripting

What You'll Need:

- A computer running **Linux** or *cough* **OSX** *cough* or **Windows 10**
  - This can also be done on Redbrick
- A Terminal Emulator
  - GNOME Terminal, Xterm, etc
- A Text Editor
  - Vim, nano, gedit etc

- No prior knowledge needed!
  - We'll bash it into you ;)

Redbrick
DCU's Networking Society

Easy Mode: Hello World

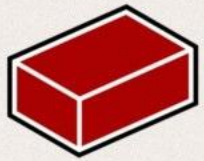
```
#!/bin/bash

FOO="Hello, World!"

echo $FOO
~
~
```

Redbrick
DCU's Networking Society

- #!/bin/bash
  - This is where your bash interpreter is located
- FOO=”Hello, World!”
  - Declares a variable ‘FOO’, assigns a string to it
- echo $FOO
  - Prints the contents of ‘FOO’
  - ‘$’ used when calling variables

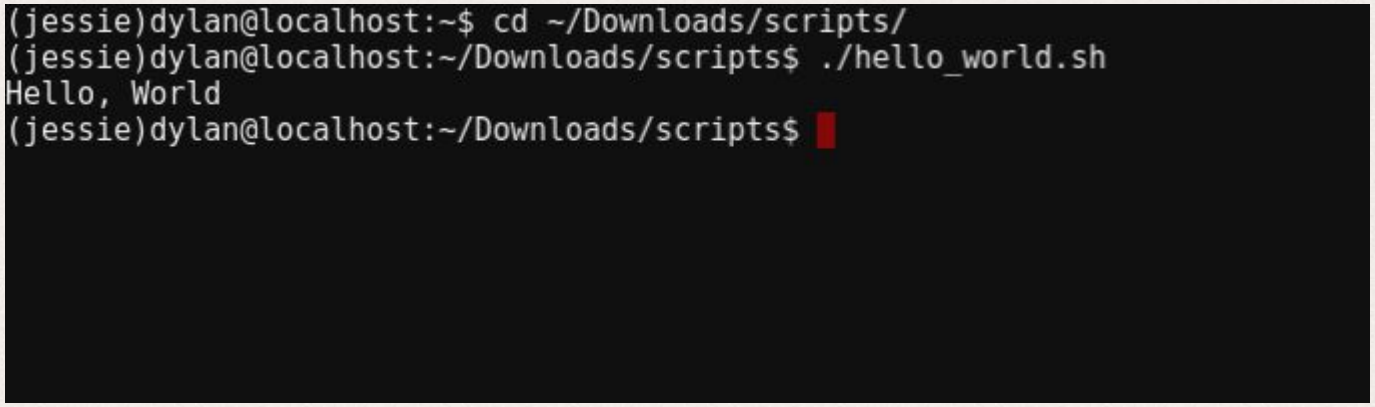
Time to make it executable

- Move to the folder your script is in:
  - cd path/to/yourscript.sh
- Make it executable:
  - chmod +x yourscript.sh
- And run:
  - ./yourscript.sh



Redbrick
DCU's Networking Society

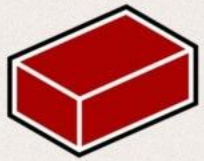
You be looking at something like this?



- You should be seeing something along these lines.
- If not, try to look really confused, we might spot you
  - or just raise your hand

Redbrick
DCU's Networking Society

A More Useful Example



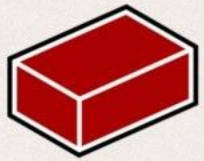
Redbrick
DCU's Networking Society

Check if a Number is Odd or Even

```
#!/bin/bash

echo "Enter a Number."
read n

num=$(expr $n % 2)
if [ $num -eq 0 ]
then
    echo "Even Number."
else
    echo "Odd Number."
fi
```

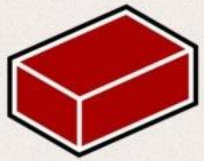
Redbrick
DCU's Networking Society

- echo "Enter The Number"
  - Prints the quoted text to the screen



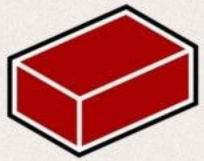
Redbrick
DCU's Networking Society

- read n
  - Allows user input, stored in variable ‘n’



Redbrick
DCU’s Networking Society

- num=$(expr $n % 2)
  - Does the calculation
  - Then saves it in the variable $num



Redbrick
DCU's Networking Society

- if [$num -eq 0]
  - Checks if this remainder is 0

- then echo “Even Number.”
  - Tells us the number is even, if the remainder is 0.

- else
  - If the remainder is not 0:

- echo “Odd Number.”
  - Prints
  - A non-zero remainder indicated an odd number

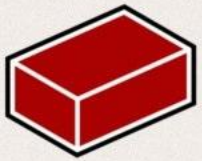
- fi
  - Terminates the ‘if’ statement.

And let’s run it!

```
(jessie)dylan@localhost:~/Downloads/college/intro_bash$ ./even_odd.sh
Enter a Number.
6
Even Number.
(jessie)dylan@localhost:~/Downloads/college/intro_bash$
```

Redbrick
DCU’s Networking Society

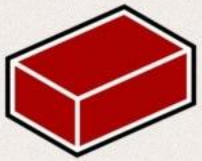
One More example!



Redbrick
DCU's Networking Society

Backing up

```
#!/bin/bash

if [ -z "$1" ]
   then
      WORK=Downloads/college
   else
      WORK=$1
fi

echo "Do you want to backup your ~/$WORK directory? (yes/no)"
read n
if [ $n = yes -o $n = y ] ; then
   if [ ! -d ~/backups ] ; then
      mkdir ~/backups
   fi
   DEST=~/backups
   NAME=work-$(date +%d%-m%-y).tar.gz
   cd ~/
   tar -zcvf $DEST/$NAME ~/$WORK
   echo "Here is your backups folder:"
   ls ~/backups
fi
```



Redbrick
DCU's Networking Society

- [-z “$1”]
  - Checks for a command line argument (CLA)
  - If there is a CLA the directory for backup will be set to it
  - Otherwise it is set to a predetermined directory

- [$n =yes -o $n =y]
  - checks for user input
  - if “yes” or “y”, script will continue
  - -o is an or statement

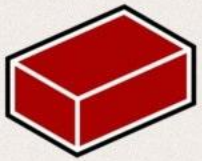
Redbrick
DCU’s Networking Society

- [! -d ~/backups] ; then
  - mkdir ~/backups
    - this checks to see if there is a directory called backups in your home directory
    - ! is not
    - -d directory
- NAME=work-$(date + %d-%m-y).tar.gz
  - this is the naming convention of our backups with the date of backup

- tar -zcvf $DEST/$NAME ~$WORK
  - is a tar command that zips the selected directory (WORK) to our backup directory

And this is the output

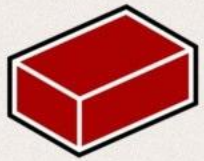
```
(jessie)dylan@localhost:~/Downloads/scripts$ ./work_backup.sh Downloads/scripts
Do you want to backup your ~/Downloads/scripts directory? (yes/no)
yes
tar: Removing leading `/' from member names
/home/dylan/Downloads/scripts/
/home/dylan/Downloads/scripts/work_backup.sh
/home/dylan/Downloads/scripts/backup.sh
/home/dylan/Downloads/scripts/hello_world.sh
Here is your backups folder:
nameoffile.tar.gz  work-081116%.tar.gz  work-081116.tar.gz  work-091116.tar.gz
(jessie)dylan@localhost:~/Downloads/scripts$
```



Redbrick
DCU's Networking Society

Bash can do a lot more...

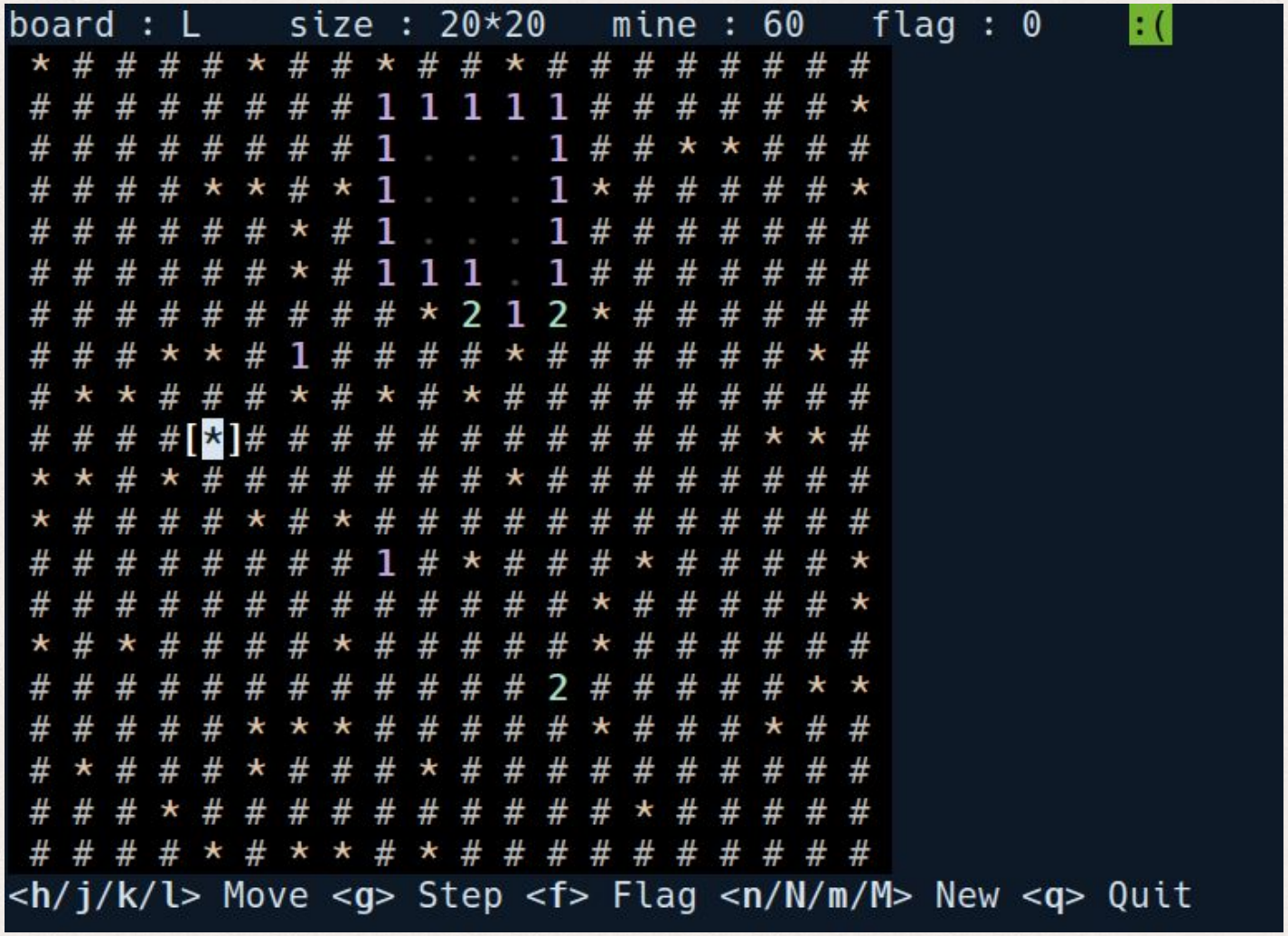
All you need is imagination!



Redbrick

DCU's Networking Society

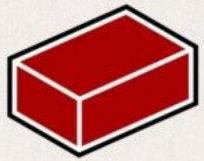
You can play Minesweeper!



Redbrick
DCU's Networking Society

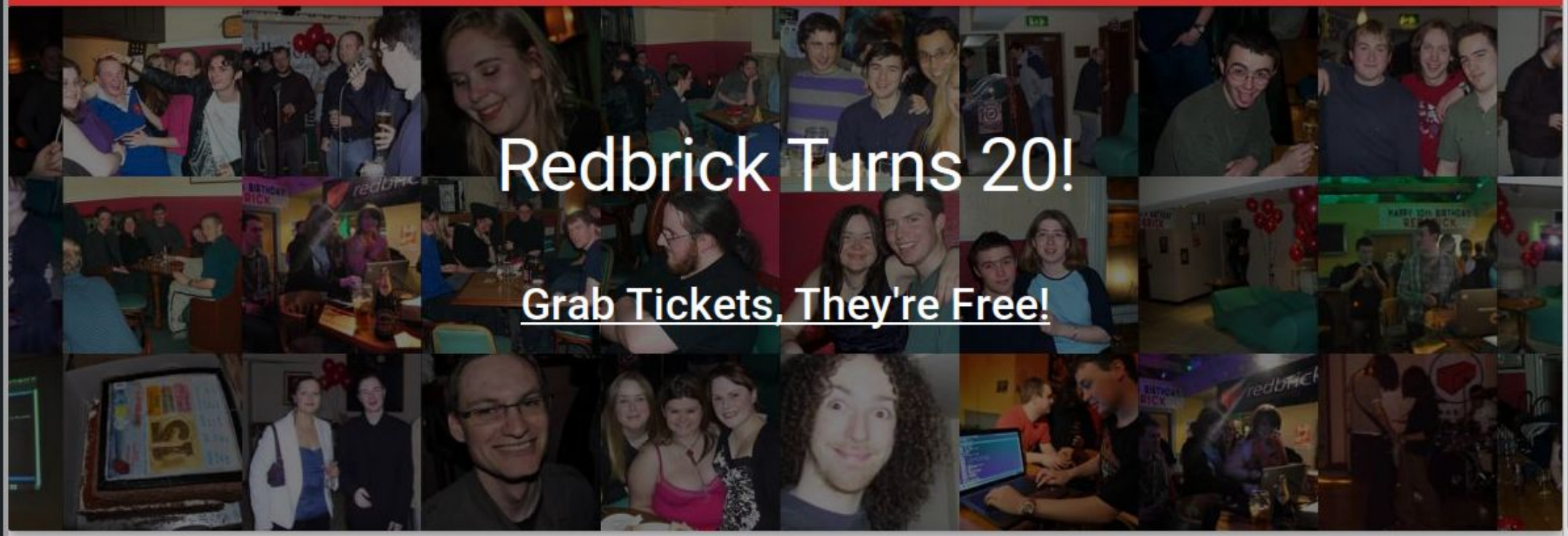
You can Edit Images!

...wait, what?



Redbrick
DCU's Networking Society

```
#!/bin/bash

icon="/home/pints/.i3/i3lock/icon.png"
tmpbg="/home/pints/tmp/screen.png"
tmp_lock="/home/pints/tmp/lock_screen.png"

xaxis=$(xdpyinfo | grep dimensions | uniq | awk '{print $2}' | cut -d 'x' -f1)
yaxis=$(xdpyinfo | grep dimensions | uniq | awk '{print $2}' | cut -d 'x' -f2)

#Grab current screen contents
scrot -z -q 100 "$tmpbg"

#Pixelate
convert "$tmpbg" -scale 10% -scale 1000% "$tmpbg"

#Tile a 10x10 circular cutout
convert -sample 10x10 xc: -draw 'circle 5,5 5,9' -negate \
        -write mpr:spot +delete \
"$tmpbg" -scale 100% -size "$xaxis"x"$yaxis" tile:mpr:spot \
+swap -compose multiply -composite "$tmp_lock"

#Add a lock icon to the centre of the image
composite -gravity center "$icon" "$tmp_lock" "$tmp_lock"

#enable i3lock with colours  modified image
i3lock --textcolor=ffffff00 --insidecolor=ffffff00 --ringcolor=ffffff00 --linecolor=ffffff00 --keyhlcolor=00FF00
80 --ringvercolor=0000FF00 --insidevercolor=00000000 --ringwrongcolor=00000055 --insidewrongcolor=FF00001c  -i "
$tmp_lock"

#clean up
rm "$tmpbg"
rm "$tmp_lock"
```



Redbrick Turns 20!

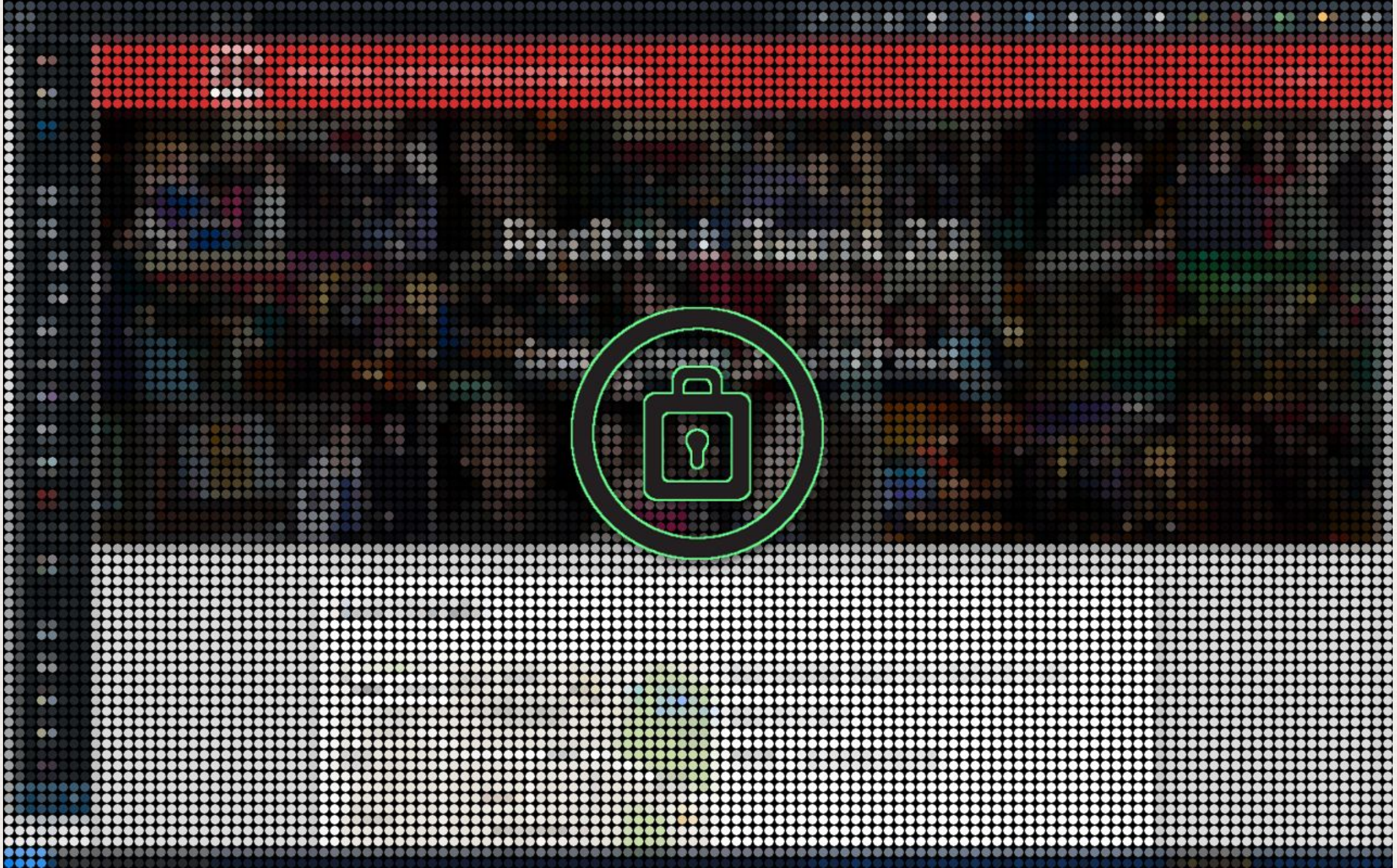
Grab Tickets, They're Free!

Event Details

Event Location: Opium Cafe, Wexford St, Dublin

Date: 9th of April

Tickets: Get Tickets From Tito



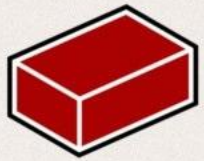
More fun awaits:

All scripts shown today are at:

- https://github.com/redbrick/HelpdeskTalks

More bash file can be found at:

- www.github.com/butlerx/bash-scripts



Redbrick

DCU's Networking Society

Questions?

Redbrick
DCU's Networking Society