

TERMINAL





SETUP

Mac

Open up the built-in terminal app

PC

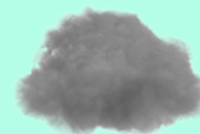
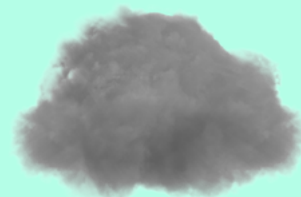
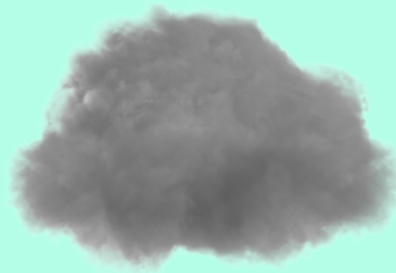
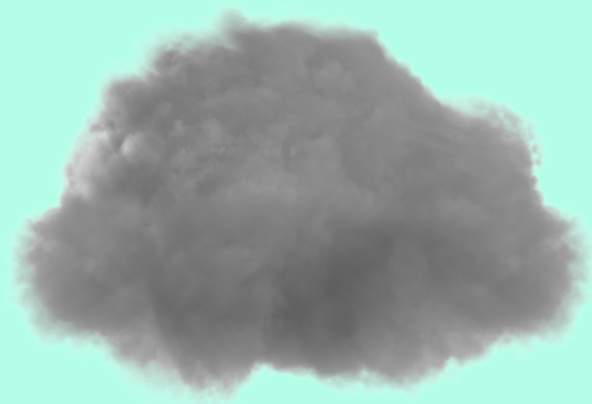
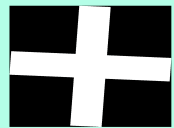
Enable Windows Subsystem
(follow the linked tutorial)



WHY

DO YOU NEED TO KNOW THIS?



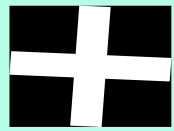


Speed!

Develop Faster

The terminal takes some getting used to, but it can be MUCH faster than using a GUI.



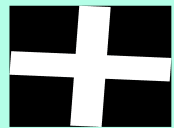
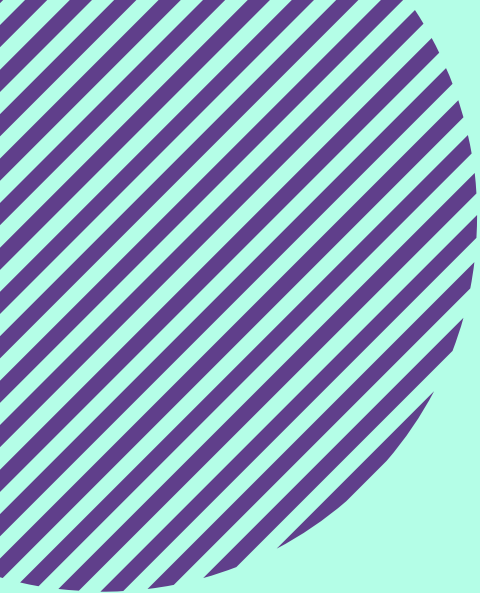


Access

With Great Power...

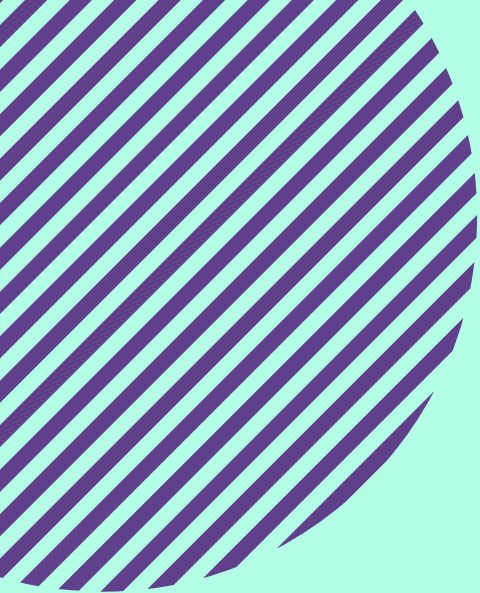
The terminal provides a "mainline" into the heart of our computer, giving us access to areas we normally don't interact with.





Tools!

Many of the tools we need are installed and used via the command line. We don't have much of a choice!



Confusing Terminology

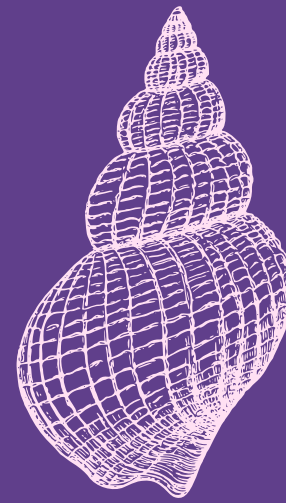
Terminal?
Shell?
Command Line?
Console?
Bash?





Terminal

A TEXT-BASED INTERFACE TO YOUR COMPUTER.
ORIGINALLY A PHYSICAL OBJECT, BUT NOW WE
USE SOFTWARE TERMINALS



Shell

THE PROGRAM RUNNING ON THE TERMINAL.

A QUICK ANALOGY!

TERMINAL

In this stupid analogy, the ATM is the terminal

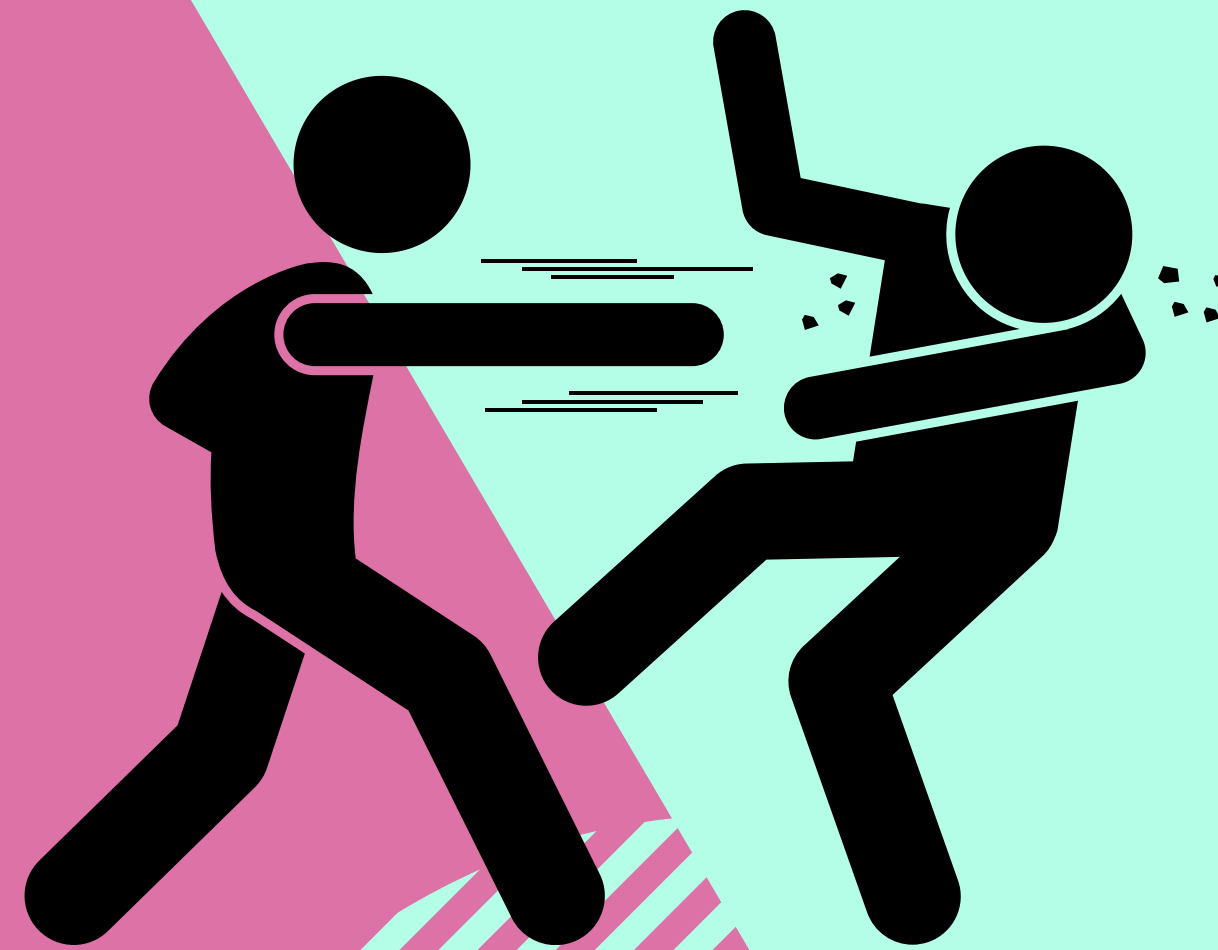
SHELL

The software running on the ATM is the shell



BASH

ONE OF THE MOST POPULAR SHELLS
(AND THE DEFAULT ON A MAC)



LS

```
TerminalYT — -bash — 61x6
[🌲 Colt-Steele: ls
Secret          app.js          llamas.py
app.css         index.html
🌲 Colt-Steele: █
```

List



Use `ls` to list the contents of your current directory

P
W
D

A screenshot of a macOS Terminal window titled "TerminalYT — -bash — 61x6". The prompt is "Colt-Steele:". The user has entered the command "pwd". The output is "/Users/RecordingUser/TerminalYT". The prompt "Colt-Steele:" is followed by a cursor.

```
Colt-Steele: pwd
/Users/RecordingUser/TerminalYT
Colt-Steele: 
```

Print Working Directory

Prints the path to the working directory
(where you currently are)

C D

```
Colt-Steele: ls
Secret          app.js          llamas.py
app.css         index.html
Colt-Steele: cd Secret/
Colt-Steele: ls
my_diary.html   passwords.txt
Colt-Steele:
```

Change Directory

Use *cd* to change and move between folders

C
D

```
TerminalYT — -bash — 61x6
[🌲 Colt-Steele: pwd
/Users/RecordingUser/TerminalYT/Secret
[🌲 Colt-Steele: cd ..
[🌲 Colt-Steele: pwd
/Users/RecordingUser/TerminalYT
[🌲 Colt-Steele: ]
```

`cd ..`

Use `cd ..` to "back up" one directory

touch

```
TerminalYT — -bash — 61x8
[🌲 Colt-Steele: ls
Secret      app.js      llamas.py
app.css     index.html
[🌲 Colt-Steele: touch purple.txt
[🌲 Colt-Steele: ls
Secret      app.js      llamas.py
app.css     index.html  purple.txt
[🌲 Colt-Steele: 
```

Touch

Use *touch* to create a file (or multiple)
Yes, the name is weird...

mkdir

```
TerminalYT — -bash — 61x9
[🌲 Colt-Steele: ls
Secret          app.js          llamas.py
app.css         index.html     purple.txt
[🌲 Colt-Steele: mkdir Plants
[🌲 Colt-Steele: ls
Plants New!      app.js          purple.txt
Secret          index.html
app.css         llamas.py
[🌲 Colt-Steele: █
```

mkdir (make directory)

mkdir will create a new directory
(or directories)

rm

```
TerminalYT — -bash — 61x9
[🌲 Colt-Steele: ls
Plants          app.js          purple.txt
Secret          index.html
app.css         llamas.py
[🌲 Colt-Steele: rm purple.txt
[🌲 Colt-Steele: ls
Plants          app.css          index.html
Secret          app.js           llamas.py
[🌲 Colt-Steele: ]
```

rm

rm will delete a file or files
It permanently removes them!

rm

```
TerminalYT — -bash — 61x9
[🌲 Colt-Steele: ls
Plants          app.css          index.html
Secret          app.js           llamas.py
[🌲 Colt-Steele: rm -rf Plants
[🌲 Colt-Steele: ls
Secret          app.js           llamas.py
app.css         index.html
[🌲 Colt-Steele: 
```

rm -rf

use *rm -rf* to delete a directory
(r = recursive, f = force)