

Using
the
command
line

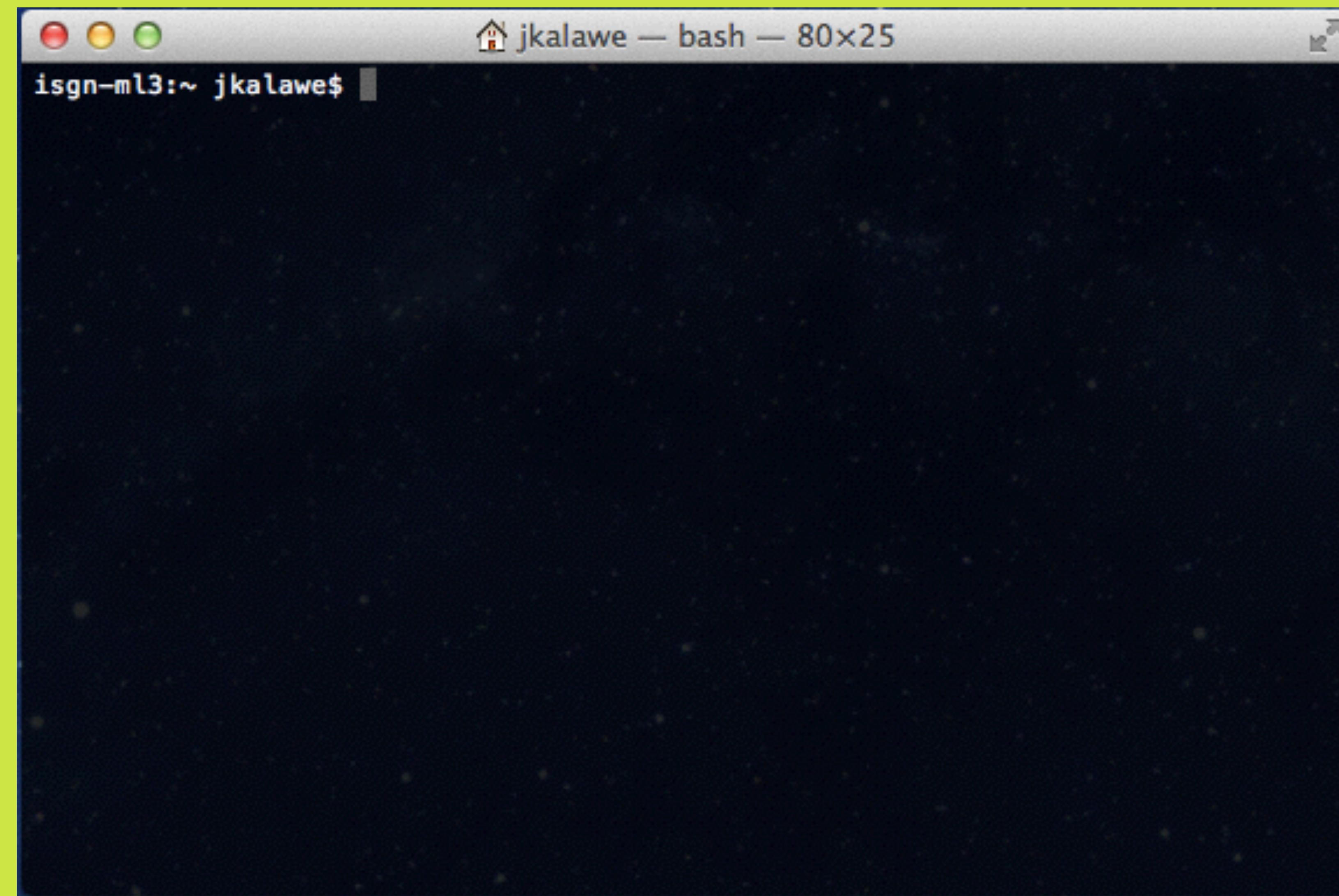


A black and white photograph of a puppy, likely a Labrador Retriever, sitting and looking up at the camera with a curious expression. The puppy is the central focus of the image.

unix is your
special friend

The TERMINAL

(don't be SCARED)





input

output

FOLLOW ALONG

commands take input to create output

```
echo
```

commands can take arguments

```
echo "this is an argument"
```

FOLLOW ALONG

OUTPUT YOUR USERNAME

```
whoami
```

OUTPUT COMPUTER UPTIME

```
uptime
```

FOLLOW ALONG

OUTPUT YOUR OWN TEXT

echo UNIX's is cool!

ESCAPE THE TEXT THIS TIME

echo “Unix’s is cool!”

Escaping Text

FOLLOW ALONG

USE QUOTES FOR TEXT

```
echo "This is my text"
```

USE BACKTICKS IF IT'S A COMMAND

```
echo "Today's date is " `date`
```

A black and white photograph of a man with glasses looking thoughtfully at a stack of books. He is wearing a light-colored shirt and a patterned tie. In the background, there are bookshelves filled with books.

the
filesystem

TERMINOLOGY

Path

Location in the filesystem of a file

Path example

/Users/yourusername

FOLLOW ALONG

OUTPUT YOUR CURRENT PATH

```
pwd
```

GO TO YOUR HOME DIRECTORY

```
cd
```

FOLLOW ALONG

LIST FILES IN THE CURRENT DIRECTORY

```
ls
```

OUTPUT A DETAILED LIST

```
ls -l
```

CREATE FILES

& DIRECTORIES

FOLLOW ALONG

MAKE a diRECTORY

```
mkdir prg-01
```

change into this diRECTORY

```
cd prg-01
```

FOLLOW ALONG

CREATE A FILE

```
touch test.txt
```

REMOVE THE FILE

```
rm test.txt
```

Redirection

TERMINOLOGY

Redirection

By default output is sent to the terminal

Redirection instead sends output to a command or file

FOLLOW ALONG

Redirect text to a file using >

```
echo "First line" > test.txt
```

Append more text to a file using >>

```
echo "Second line" >> test.txt
```

FOLLOW ALONG

view the contents of a file

```
cat test.txt
```



our first
program

TERMINOLOGY

COMPUTER PROGRAM

Just a bunch of commands in a text file

FOLLOW ALONG

OUR PROGRAM WILL

OUTPUT TEXT

TO A FILE

FOLLOW ALONG

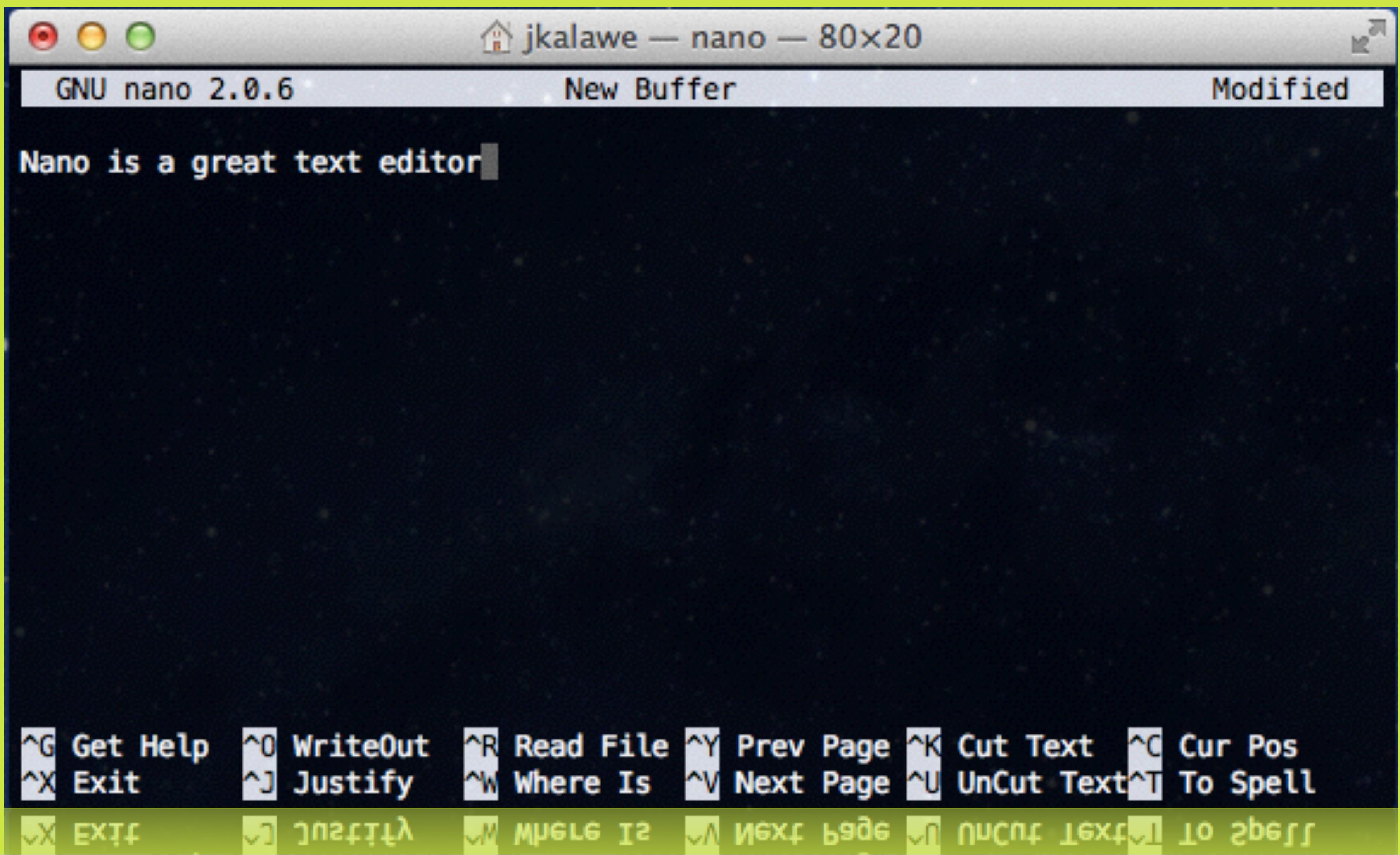
LET'S MAKE A PROJECT DIRECTORY

```
mkdir ~/Documents/Coding100
```

GO TO OUR PROJECT DIRECTORY

```
cd ~/Documents/Coding100
```

nano is a text editor



FOLLOW ALONG

LET'S WRITE OUR PROGRAM USING nano

```
nano hello.bash
```

FOLLOW ALONG

Identify the Language We're Using

```
#!/bin/bash
```

WRITE TEXT TO A TEXT FILE

```
echo 'hello world' > hello.txt
```

FOLLOW ALONG

WRiTE another Line

```
echo 'hello again' >> hello.txt
```

LET'S SAVE OUR file

```
type 'control' and 'o'
```

FOLLOW ALONG

Exit nano

type ‘control’ and ‘x’

TEST OUR
PROGRAM

FOLLOW ALONG

Run the program using bash

```
bash hello.bash
```

FOLLOW ALONG

check if our program wrote the file

```
ls
```

check contents of file

```
nano hello.txt
```