Intro to Programming
for Information and Data Science

INFX 598i/j

Joel Ross
Spring 2017

Hi, I'm Joel!

About Joel

Senior Lecturer

Email joelross@uw.edu **Office** MGH 330C

Who are you?

Ask the person across from you...

- Who are you?
- Where did you grow up?
- **8:30am**: too early, or way too early?
- What is one thing you're hoping to learn from this class (what is your "learning objective")?

What are we doing here?

Today's Objectives

By the end of the class you should be able to

- Understand the purpose and structure of **computer programming**
- Navigate the **course** materials
- Control a computer using the **command-line**
- Reflect on how to **solve problems** (if time)

Information School

UNIVERSITY of WASHINGTON

WE MAKE INFORMATION WORK

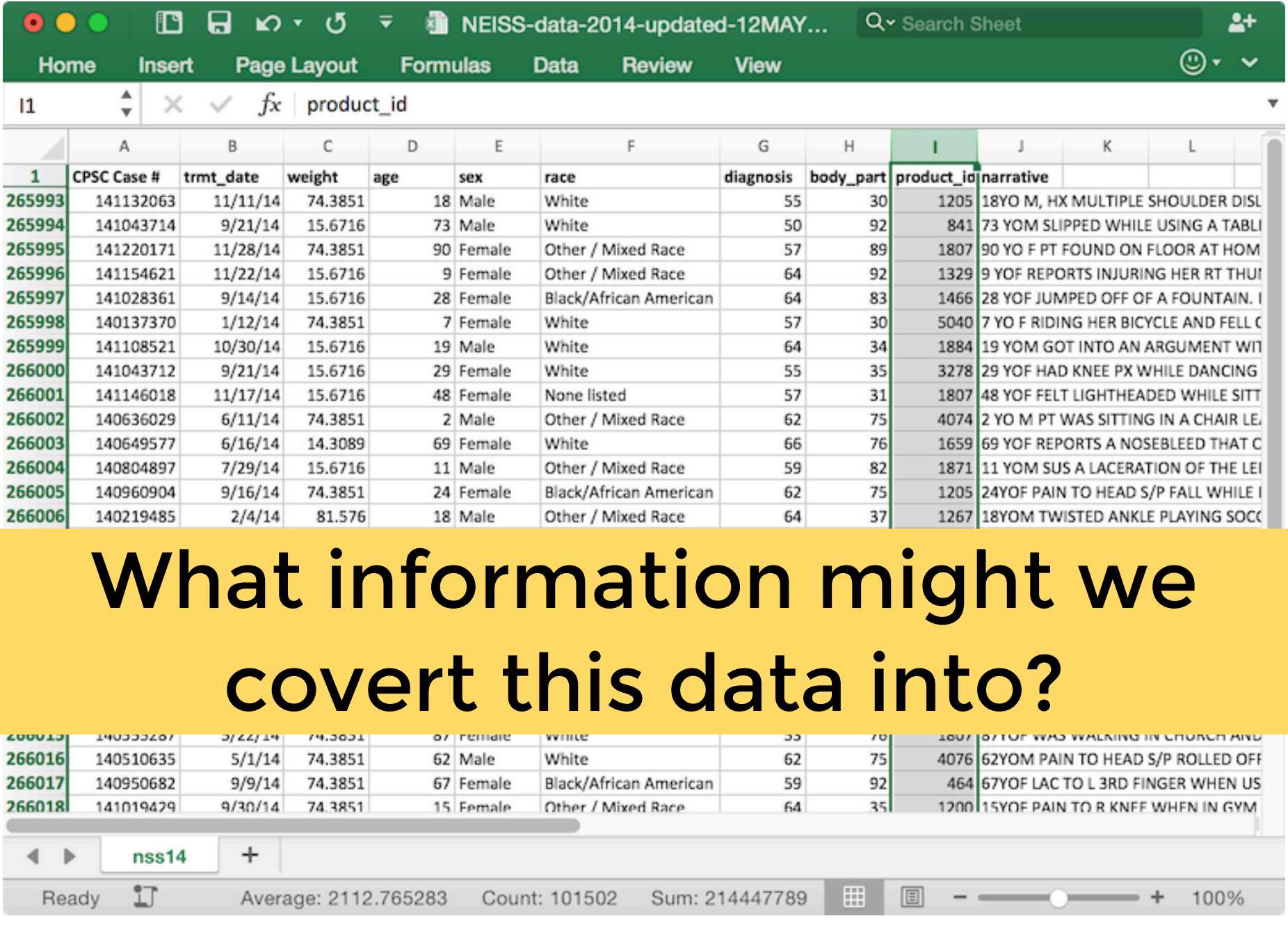
We need to convert

data ← raw numbers

into

information

↑ interpreted data



	A	B	C	D	E	F	G	H	I	J
1	CPSC Case #	trmt_date	weight	age	sex	race	diagnosis	body_part	product_id	narrative
265993	141132063	11/11/14	74.3851	18	Male	White	55	30	1205	18YO M, HX MULTIPLE SHOULDER DISL
265994	141043714	9/21/14	15.6716	73	Male	White	50	92	841	73 YOM SLIPPED WHILE USING A TABLI
265995	141220171	11/28/14	74.3851	90	Female	Other / Mixed Race	57	89	1807	90 YO F PT FOUND ON FLOOR AT HOM
265996	141154621	11/22/14	15.6716	9	Female	Other / Mixed Race	64	92	1329	9 YOF REPORTS INJURING HER RT THUI
265997	141028361	9/14/14	15.6716	28	Female	Black/African American	64	83	1466	28 YOF JUMPED OFF OF A FOUNTAIN. I
265998	140137370	1/12/14	74.3851	7	Female	White	57	30	5040	7 YO F RIDING HER BICYCLE AND FELL C
265999	141108521	10/30/14	15.6716	19	Male	White	64	34	1884	19 YOM GOT INTO AN ARGUMENT WIT
266000	141043712	9/21/14	15.6716	29	Female	White	55	35	3278	29 YOF HAD KNEE PX WHILE DANCING
266001	141146018	11/17/14	15.6716	48	Female	None listed	57	31	1807	48 YOF FELT LIGHTHEADED WHILE SITT
266002	140636029	6/11/14	74.3851	2	Male	Other / Mixed Race	62	75	4074	2 YO M PT WAS SITTING IN A CHAIR LE
266003	140649577	6/16/14	14.3089	69	Female	White	66	76	1659	69 YOF REPORTS A NOSEBLEED THAT C
266004	140804897	7/29/14	15.6716	11	Male	Other / Mixed Race	59	82	1871	11 YOM SUS A LACERATION OF THE LEI
266005	140960904	9/16/14	74.3851	24	Female	Black/African American	62	75	1205	24YOF PAIN TO HEAD S/P FALL WHILE I
266006	140219485	2/4/14	81.576	18	Male	Other / Mixed Race	64	37	1267	18YOM TWISTED ANKLE PLAYING SOCC
266016	140510635	5/1/14	74.3851	62	Male	White	62	75	4076	62YOM PAIN TO HEAD S/P ROLLED OFF
266017	140950682	9/9/14	74.3851	67	Female	Black/African American	59	92	464	67YOF LAC TO L 3RD FINGER WHEN US
266018	141019429	9/30/14	74.3851	15	Female	Other / Mixed Race	64	35	1200	15YOF PAIN TO R KNEE WHEN IN GYM

What information might we covert this data into?

Should we do this by hand?

Slow Tedious Error-prone

How to *scale*? How to *reproduce*?

Solution: get someone else to do the boring work for us!

Our new assistant

We need to tell the computer what to do!

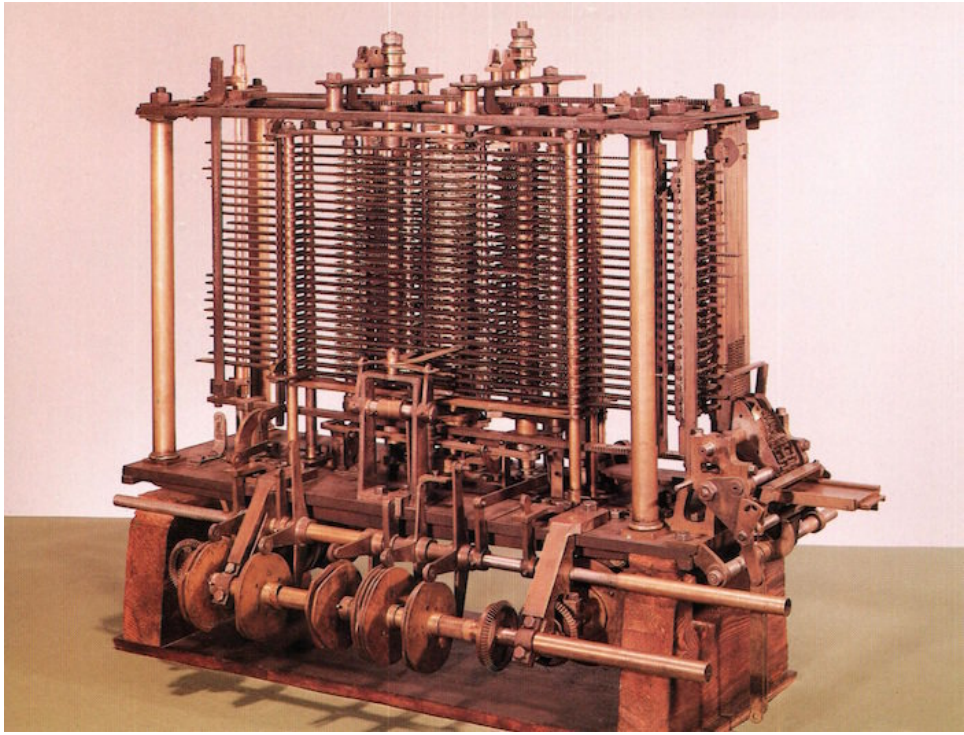
Problem

Computers don't speak English!

Programming

Writing instructions for a computer in a language it understands

First Programmer: Ada Lovelace (1815-1852)



Charles Babbage's **Analytical Engine**
(designed 1837; never built)

Binary

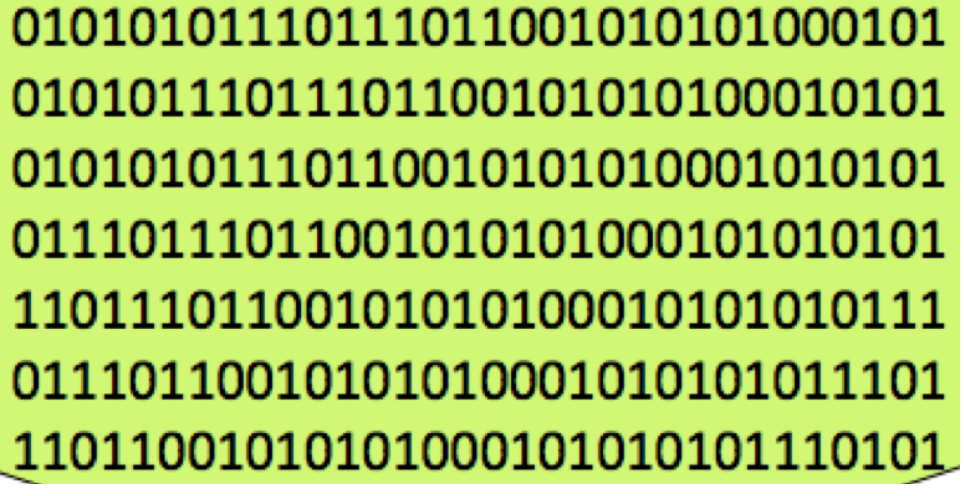
01010101110111011001010101000101	LOAD Contents of A
01010111011101100101010100010101	LOAD Contents of B
01010101110110010101010001010101	LOAD Contents of C
01110111011001010101000101010101	MULTIPLY C and 4 and STORE in TMP
11011101100101010100010101010111	SUBTRACT TMP from B and STORE in TMP
01110110010101010001010101011101	ADD A to TEMP and STORE in TEMP
11011001010101000101010101110101	STORE TEMP in D

Interpreter

```
# Write this instead #
LOAD A
LOAD B
LOAD C
MULTIPLY C and 4 and STORE in TMP
SUBTRACT TMP from B and STORE in TMP
ADD A to TEMP and STORE in TEMP
STORE TEMP in D
```

Have a computer program do a find-and-replace to change `LOAD` to `010101`, etc.

Abstraction



LOAD Contents of A
LOAD Contents of B
LOAD Contents of C
MULTIPLY C and 4 and STORE in TMP
SUBTRACT TMP from B and STORE in TMP
ADD A to TEMP and STORE in TEMP
STORE TEMP in D

```
D = A + (B - 4*C)
```

Abstraction

The process of *generalization*; of working with higher-level representations rather than specific details

Programming Language

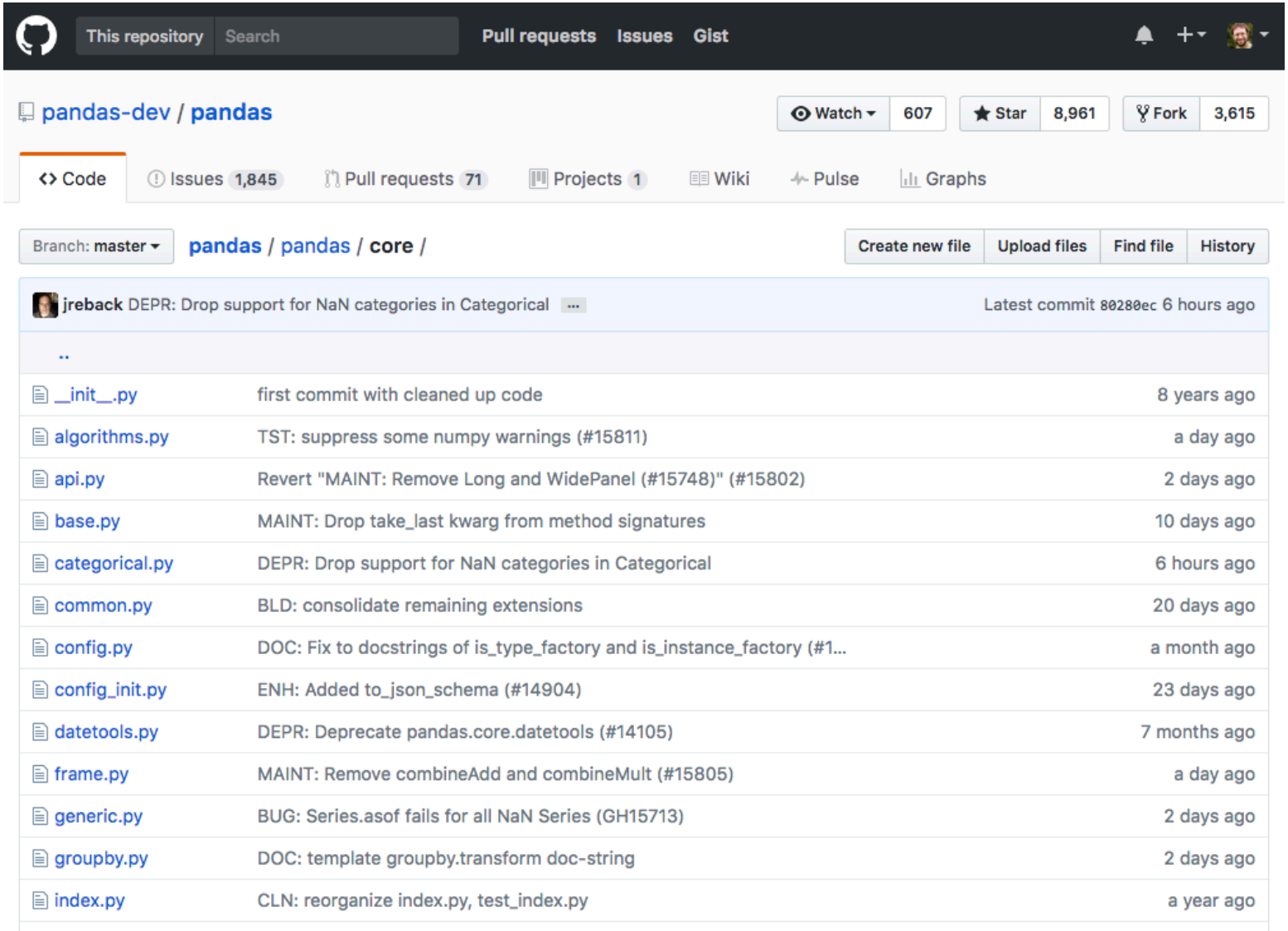
A language that a human can write, and which can be interpreted by a computer

Writing Code

```
# This is the programming language Python
# It means something reasonable, I swear!

import math

def hypot(a, b, c):
    det = math.sqrt(b*b - 4*a*c)
    x1 = (-b + det)/(2*a)
    x2 = (-b - det)/(2*a)
    return (x1,x2)

x = hypot(1,5,-14)
print(x)

# Any idea what this program does?
```

Different languages have the same basic structure (but different grammar)

Managing Code



Course Objective

Learn to give instructions to computers so they do the boring stuff!

Algorithm

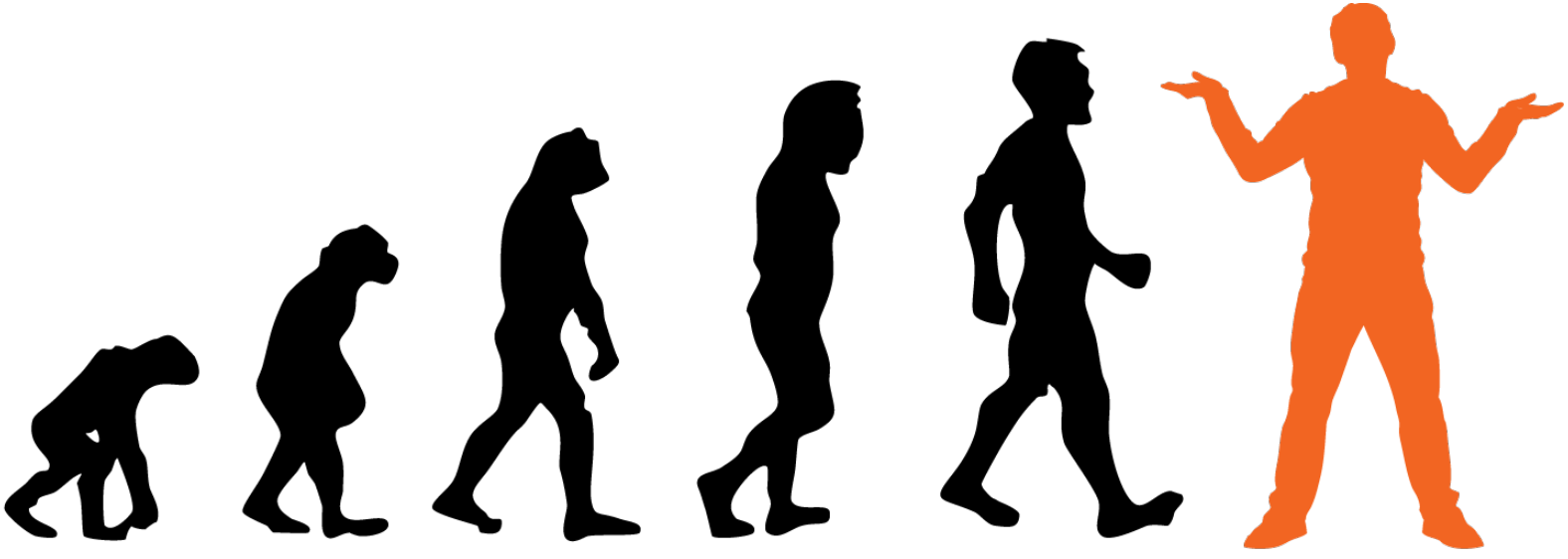
A set of step-by-step instructions for solving a problem.

Needs to be *unambiguous, executable,* and *terminating*

Programming involves **a lot** of failure and frustration

Course Resources

Canvas has the **syllabus** and **assignment** details.



Learning to Learn

Learning Modules

Course Resources

Canvas has the **syllabus** and **assignment** details.

GitHub has all the code (including **assignments** and **learning modules**)

Slack is how you will **ask questions**, **collaborate**, and see **announcements**.

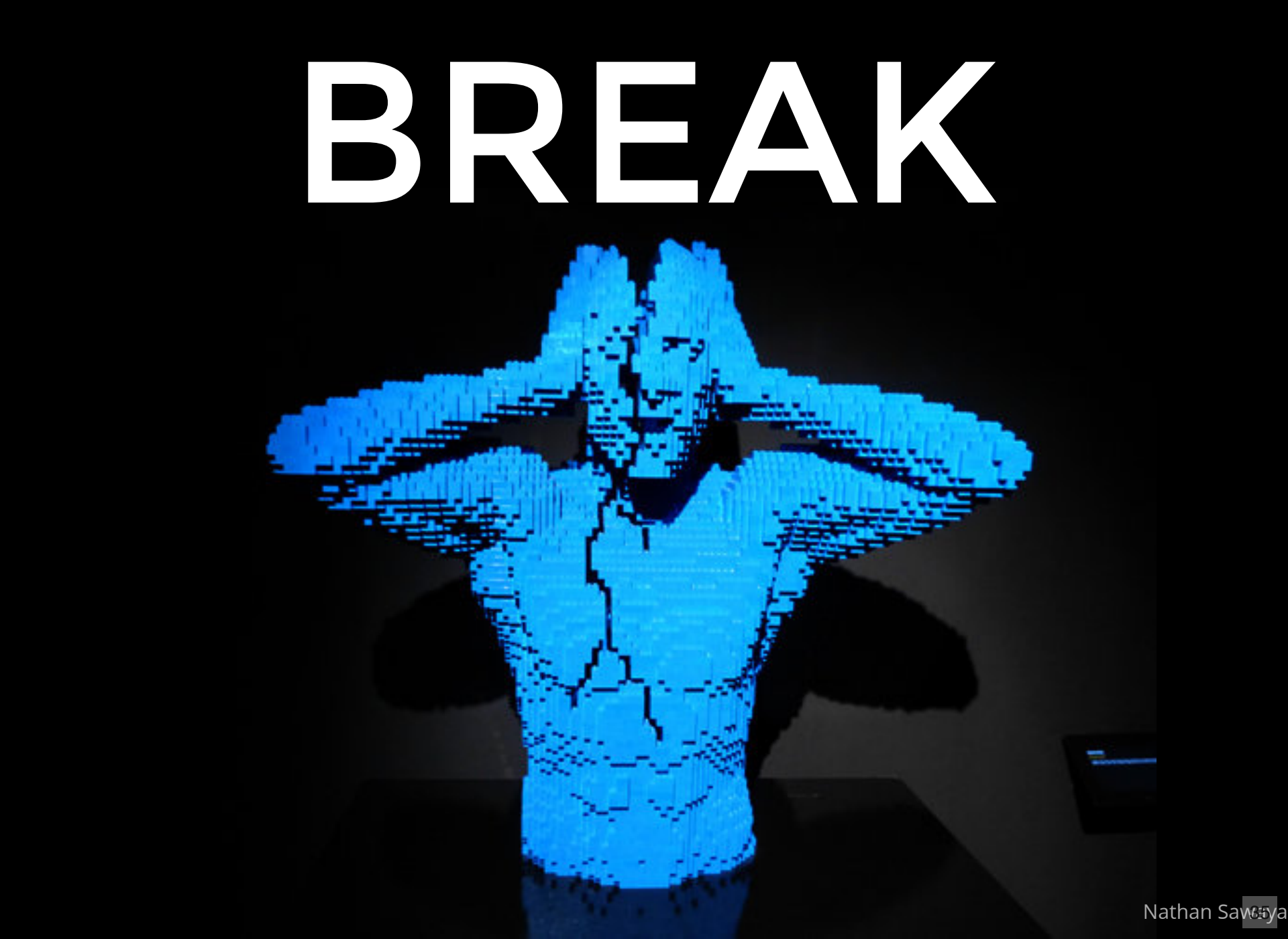
Intro to Programming

for Information and Data Science

INFX 598i/j

Any questions so far?

BREAK



Command Line

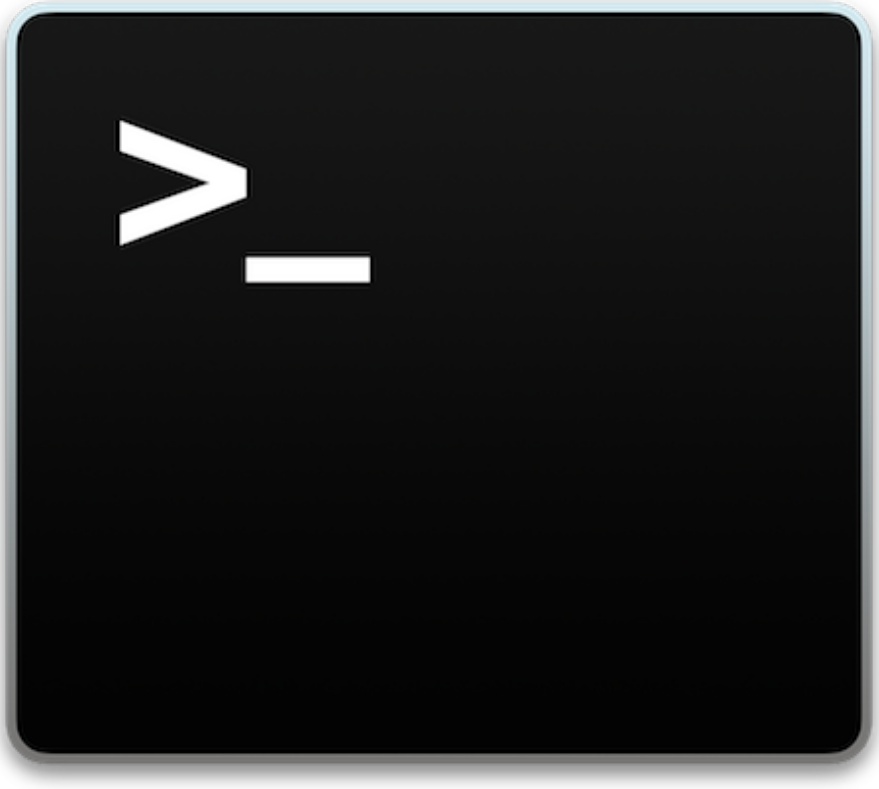
Graphical User Interface (GUI)

- Windows, Icons, Menus, Pointers (e.g,. WIMP)
- Easy to learn
- Not much more efficient for experts

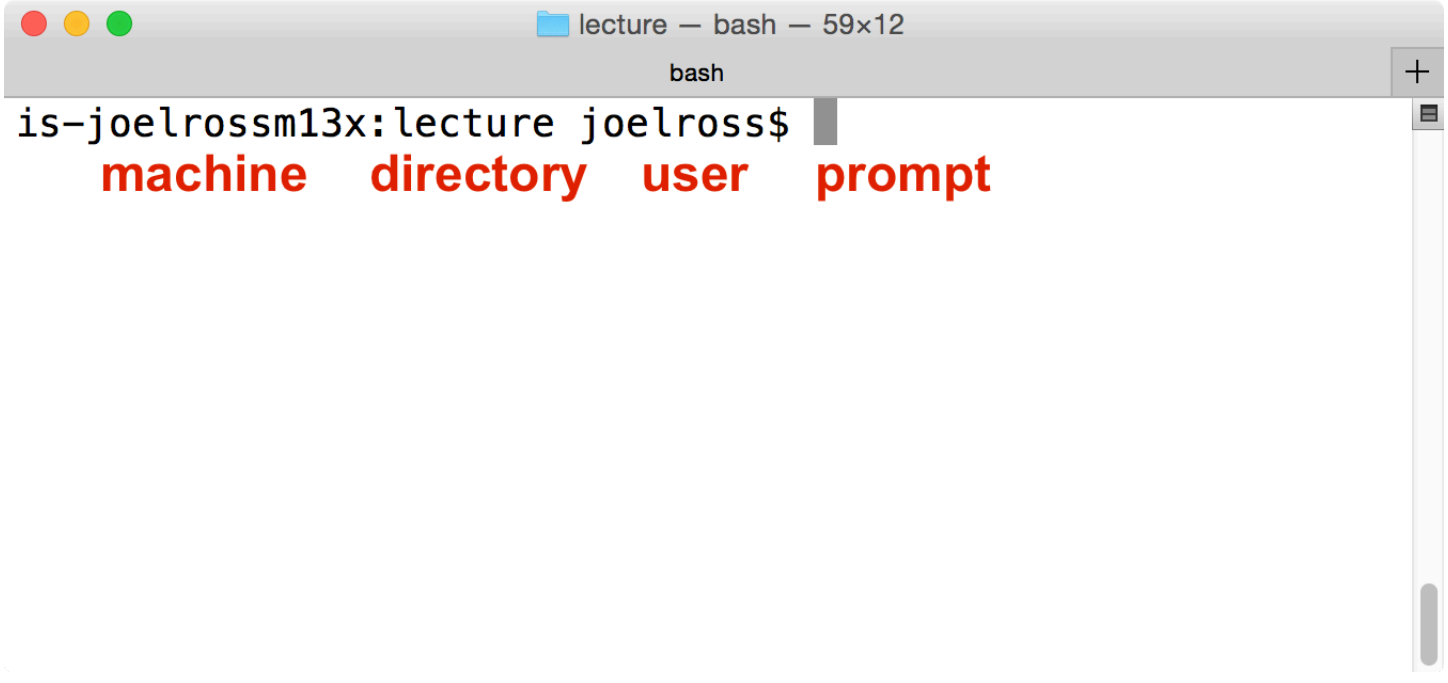
Command Line Interface

- Text-based interface
- Hard to learn
- Very efficient for experts
- Like programming!

Command Shell



(command + space, search for "Terminal")

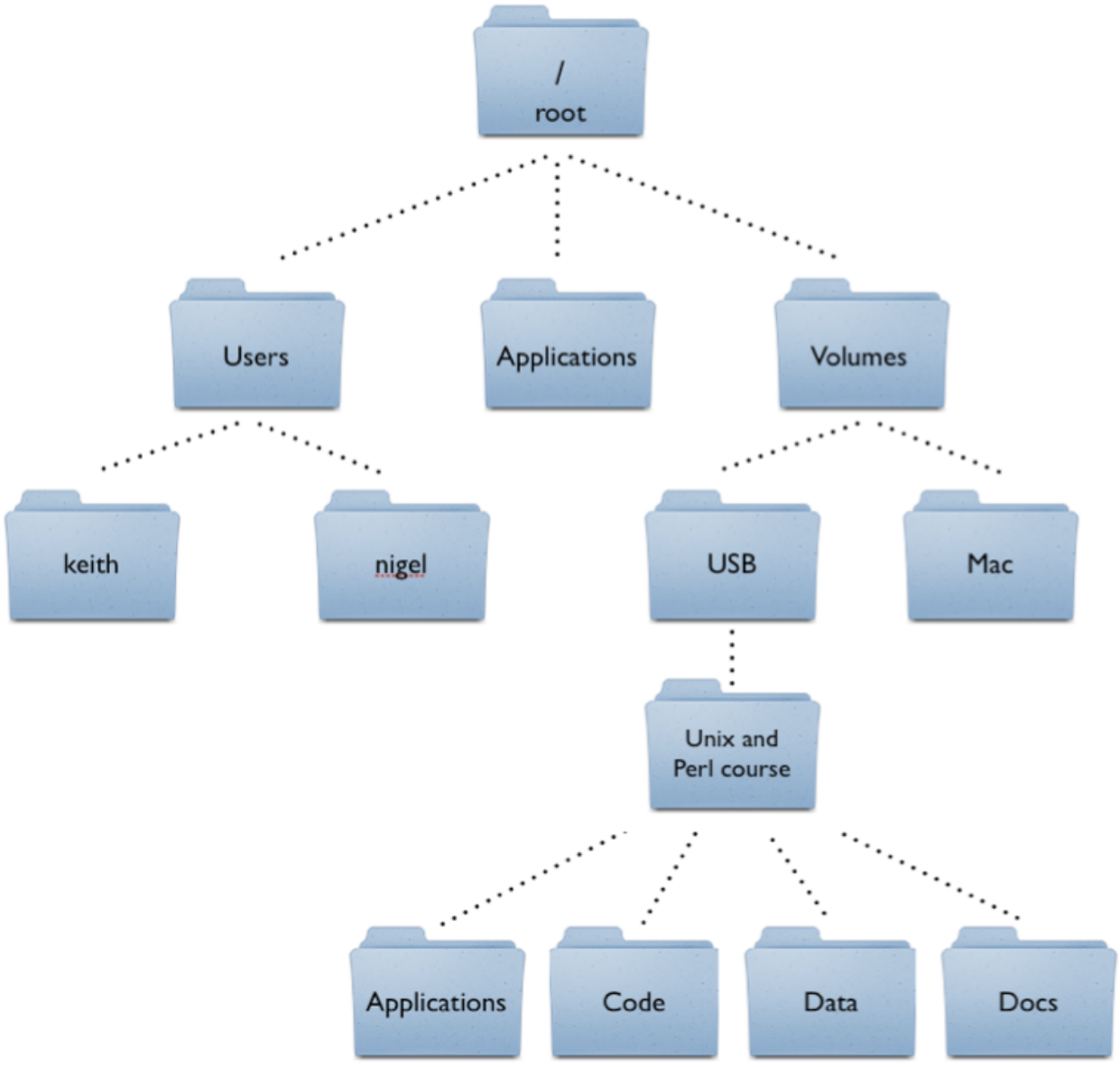

lecture — bash — 59×12
bash
is-joelrossm13x:lecture joelross$
machine
directory
user
prompt

Where are we?

```
pwd
```

(**p**rint **w**orking **d**irectory)

Directory Tree



Change Location

```
cd folder
```

command

argument
(required)

(**c**hange **d**irectory)

What's here?

```
ls [folder]
```

optional argument
(don't type the brackets)

(list contents)

Paths

`/absolute/path/to/file`

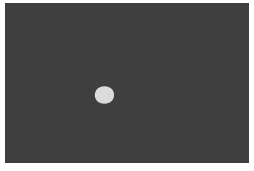
leading slash

How to get there *starting from the root*

`relative/path/to/file`

How to get there *starting from here*

Path Symbols



(a period):
the *current* directory

(two periods):
the *parent* directory

Path Practice

```
# if I start here:
$ pwd
/Users/iguest/Desktop

# and then do this:
$ cd /../Desktop

#where do I end up?
```

ALWAYS USE RELATIVE PATHS IN CODE!

More File Commands

`mkdir` (**m**ake **dir**ectory)

`rm` (**r**e**m**ove file or folder)

`cp` (**c**o**p**y file or folder)

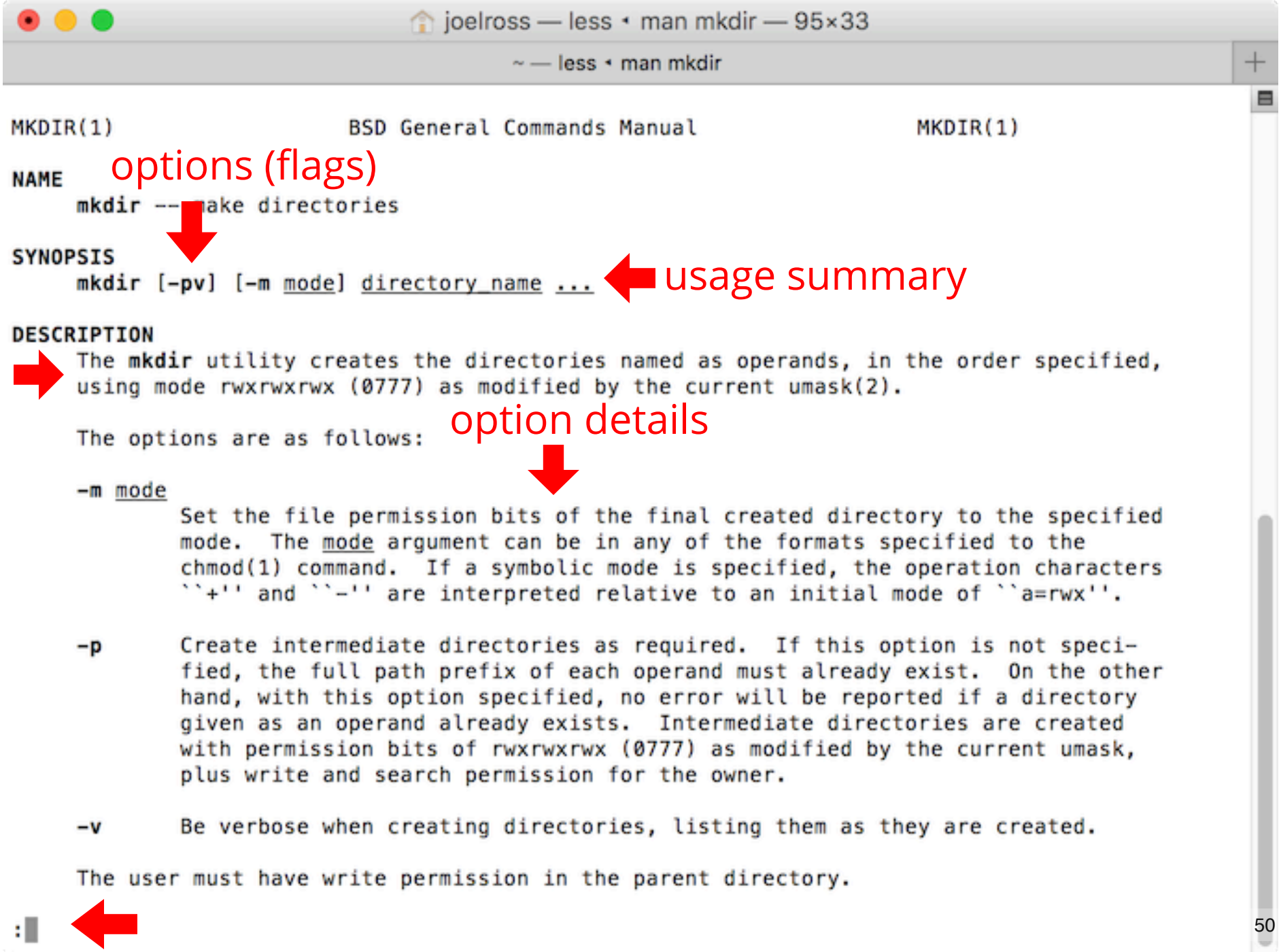
`open` (Mac: open a file or folder)

`start` (Windows: open a file or folder)

Check the Manual

```
man command
```

(**man**ual)


joelross — less • man mkdir — 95×33
~ — less • man mkdir
MKDIR(1) BSD General Commands Manual MKDIR(1)
options (flags)
NAME
mkdir -- make directories
SYNOPSIS
mkdir [-pv] [-m mode] directory_name ...
usage summary
DESCRIPTION
The mkdir utility creates the directories named as operands, in the order specified, using mode rwxrwxrwx (0777) as modified by the current umask(2).
option details
The options are as follows:
-m mode
Set the file permission bits of the final created directory to the specified mode. The mode argument can be in any of the formats specified to the chmod(1) command. If a symbolic mode is specified, the operation characters ``+'' and ``-'' are interpreted relative to an initial mode of ``a=rwx''.
-p
Create intermediate directories as required. If this option is not specified, the full path prefix of each operand must already exist. On the other hand, with this option specified, no error will be reported if a directory given as an operand already exists. Intermediate directories are created with permission bits of rwxrwxrwx (0777) as modified by the current umask, plus write and search permission for the owner.
-v
Be verbose when creating directories, listing them as they are created.
The user must have write permission in the parent directory.
:
50

Display Text

```
echo "message"
```

(**echo** text back)

When in doubt:

ctrl-c (**control** and **c**)

to cancel!

Redirects

`>`

Put output in file instead of display

```
echo "Hello World" > hello.txt
```

`>>`

Append to end of file

```
echo "Goodbye :)" >> hello.txt
```

`<`

Take input from file (less common)

```
cat < hello.txt
```

`|`

Take output and "pipe" (send) to next command

```
cat hello.txt | wc
```

Visual Studio Code

(command + space, search for "Visual Studio")

Module 2 exercise-1

Command Reference

Action	Syntax
Copy a file	`cp OLD_FILE NEW_FILE`
Move a file	`mv OLD_FILE NEW_FILE`
Delete a file (careful!)	`rm FILE`
Open a file	`open FILE` [Mac]; `start FILE` [Windows]
View file contents on command-line	`less FILE`
Output file contents	`cat FILE`
See previous commands executed	`history`

Action	Syntax
Make your computer speak [Mac]	`say "Text to say"`
Do the same thing again	`!!`
Watch Star Wars	`telnet towel.blinkenlights.nl` (use `ctrl-]` then `quit` to exit)

Efficient CLI Usage

- Use `tab` to automatically fill in the names of files and folders

- Use `up/down` arrows to access previously entered commands


THIS IS
PROGRAMMING!
imgflip.com

Shell Script

A list of commands (in order) that we want to run as a bunch; the "script" that our program should follow.

Make a Shell Script

What we did...

- Introduced programming as a concept
- Learned some common command-line utilities
- Wrote a simple shell script!

Action Items!

- Be comfortable with **modules 0-3** by Thurs
  - Install software per **module 1**
- Recommendation: don't start until Assignment 1 until Thursday!

Thursday: git and GitHub