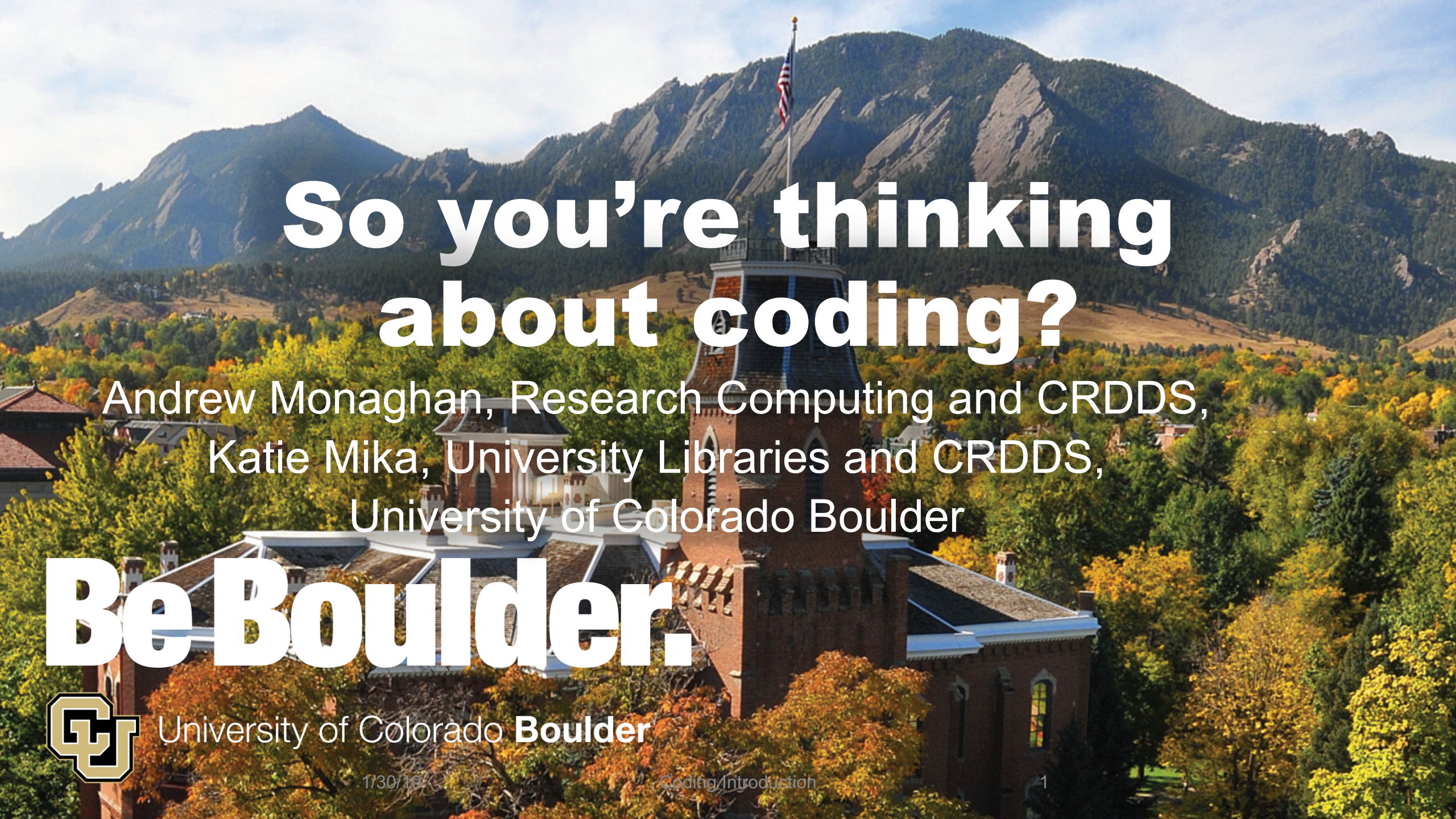


So you're thinking about coding?

Andrew Monaghan, Research Computing and CRDDS,
Katie Mika, University Libraries and CRDDS,
University of Colorado Boulder

Be Boulder.

University of Colorado **Boulder**

What is coding?

- **Coding** is the process of encoding instructions for execution by a computer. Coding results in software.
- **Coding** is synonymous with programming.
- **Coding** is done in a variety of programming languages.

Who codes, and why?

- Everyone! For example:
  - Historians mining ancient texts.
  - Psychologists simulating human behavior.
  - Political scientists studying voting patterns.
  - Chemists designing new compounds.
  - Biologists deciphering animal or plant genomes.
  - Neuroscientists exploring brain function.
  - Engineers designing nanomaterials.
  - Geologists simulating earthquakes.
  - Meteorologists modeling climate.

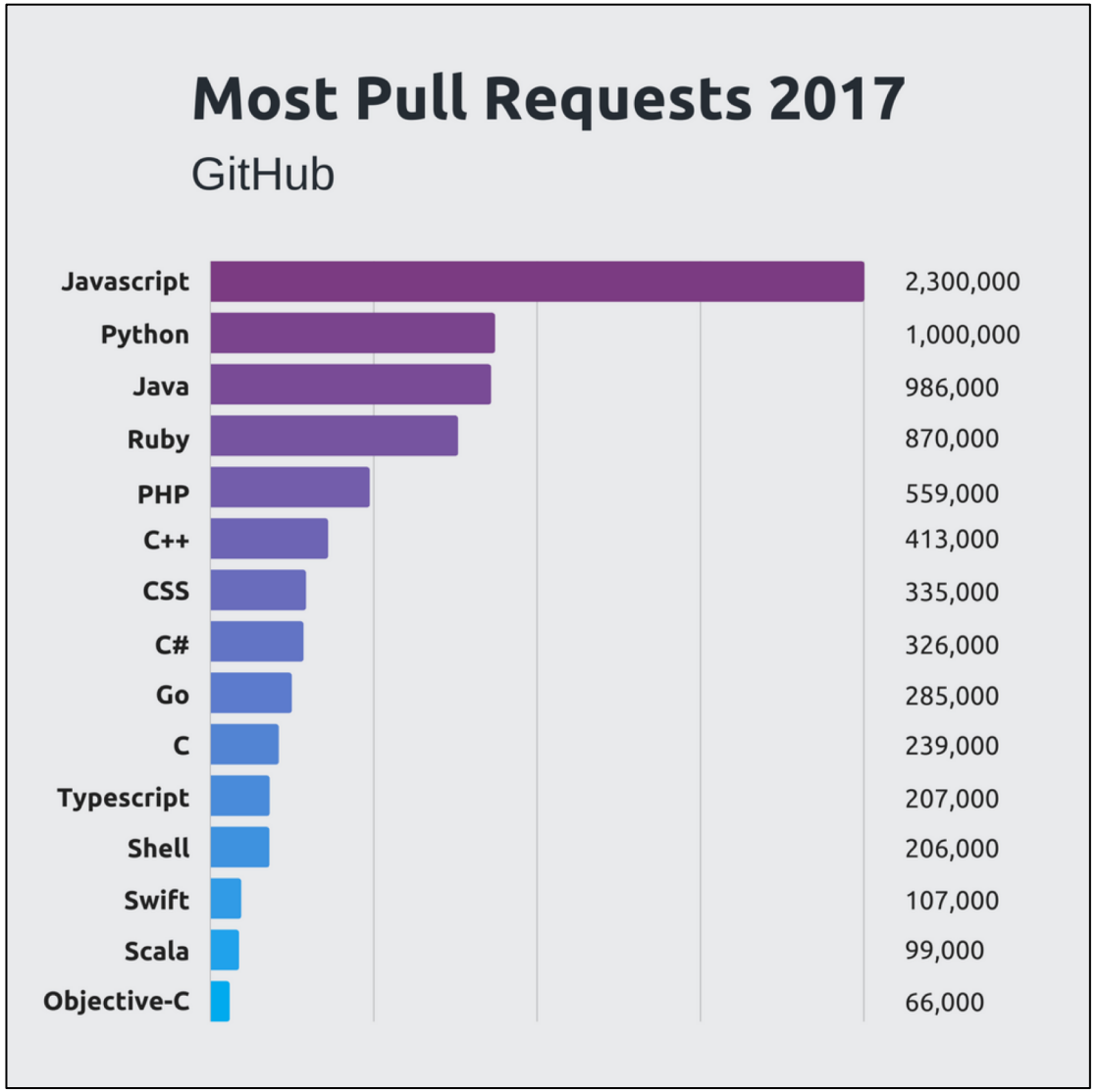
Programming languages: The 5 generations

- 1GL: Machine language
- 2GL: Assembly language
- 3GL: Imperative language
- 4GL: Declarative language
- 5GL: Natural language

Nearly all of the coding we presently do falls into these two categories

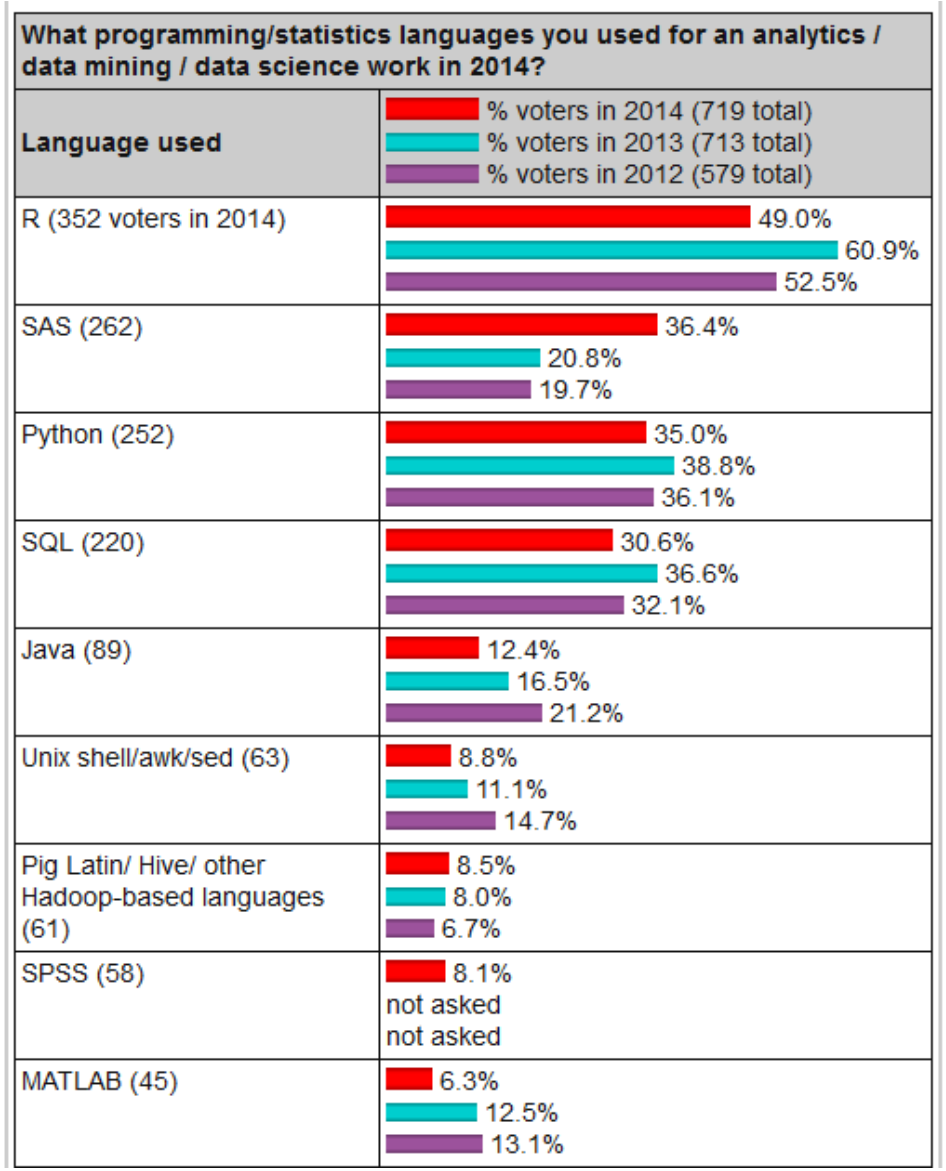
https://en.wikipedia.org/wiki/Programming_language

Most used languages, all categories



https://octoverse.github.com/

Most used languages, data science



http://bigdata-madesimple.com

Some commonly used languages in the sciences

- Fortran, C, C++ (“compiled languages”)
- R, Python, Perl, Java, Matlab (“interpreted languages”)
- Bash, C-shell (shell languages)

Some common aspects of all languages

- Variables
- Arithmetic expansion
- Decisions
- Loops
- Arguments
- Functions

…Examples of each follow using the *Bash* shell language.

Common aspect #1: Variables

A variable can contain a number, a character, a string of characters.

- Variables enable commonly used values to be recorded only once.
- Variables enable arrays of data to be represented more easily.
- Variables enable values that change frequently to be coded only once.

```
$ PI=3.14159
$ name=(Andy Monaghan)
$ echo "${name[0]} loves $PI"
$ Andy loves 3.14159
```

Common aspect #2: Arithmetic Expansion

Arithmetic expansion provides a mechanism for evaluating an arithmetic expression and substituting its value. For example:

```
$ sqr_two=$(( 2 * 2 ))
$ echo ${sqr_two}
$ 4
```

Common aspect #3: Decisions

Decisions enable conditional expressions to be evaluated.

- The `if` statement is the most common. For example:

```
x=$(date +%M) #query minutes after the hour

if [ $x -gt 30 ] ; then
  echo "last half of the hour"
elif [ $x -lt 15 ] ; then
   echo "first quarter of the hour"
else
   echo "we're at ${x}"
fi
```

Common aspect #4: Loops

Loops enable operations to be done on arrays of data. For example:

```
list="a b c"

for v in ${list} ; do
   echo $v
done
```

Common aspect #5: Arguments

Arguments allow variables to be passed into programs. For example:

----- start program calcsine.sh ---

```
#!/bin/bash
# Calculate the sine of the argument.

sine=$(echo "s($1)" | bc -l )
echo "The sine of $1 is ${sine}"
```

----- end program ------

Now run the program with an argument:

```
> bash calcsine.sh 3.2
```

```
The sine of 3.2 is -.058374143427579909 13
```

Common aspect #6: Functions

Functions enable repetitive operations to be coded once and then reused. For example:

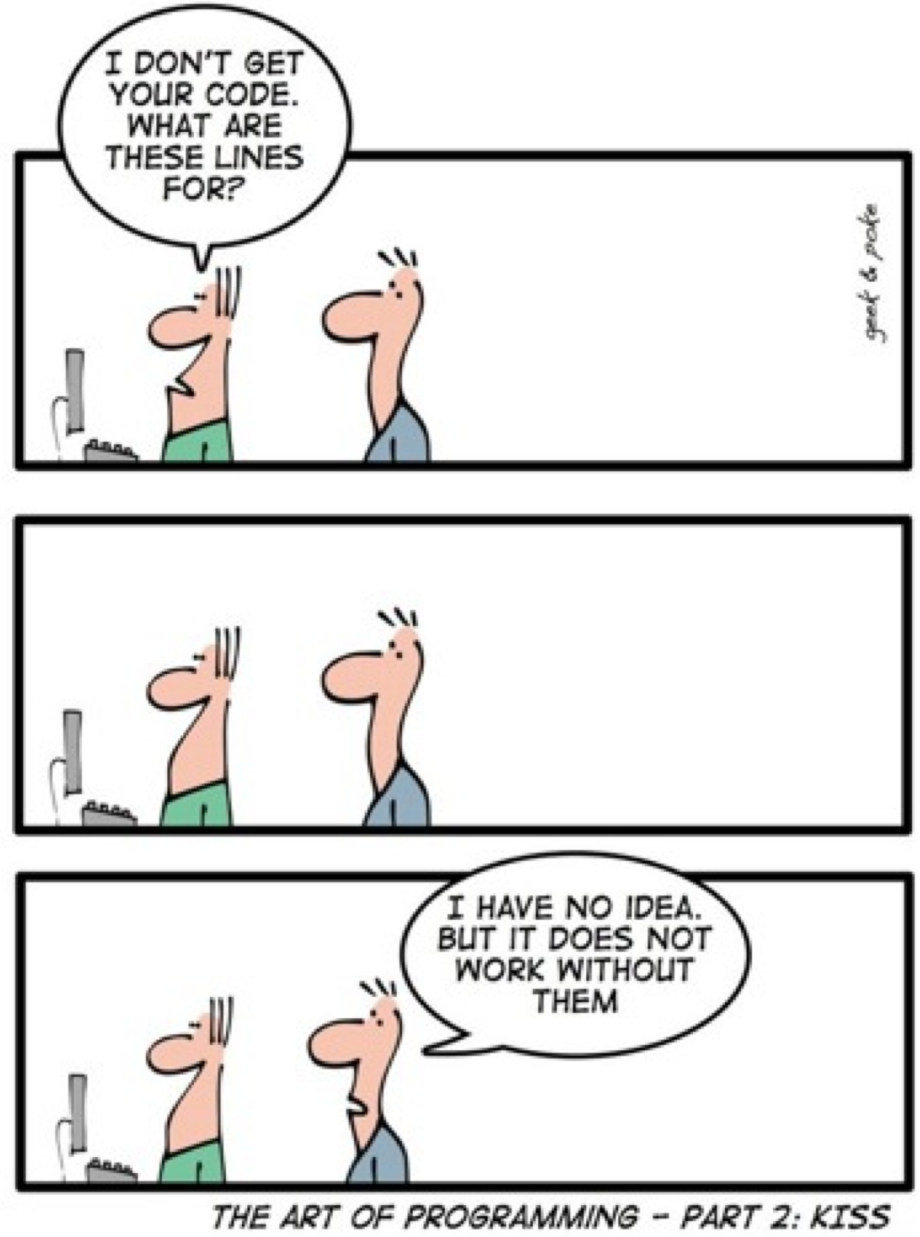
```
----start program-----
#!/bin/bash
#create a function that echoes back the input

gimme () {
      echo $1
}

#now test
gimme Hello
gimme World
```

Coding best practices

- Make a plan (write out beforehand)
- Document code (comments, README)
- Styling (group, indent, capitalize)
- Use functions where possible
- Avoid ‘hardcoding’ (use variables)
- Version control (Github)



Programming with Python

- Python is an interpreted language*.
- Wide range of uses, considered “general purpose”.
- It is open source, intuitive and has thousands of packages.
- Development environments: Jupyter, console, etc.

- Let’s do a Mini tutorial: Using Python for human language interpretation

Programming with R

- R is an interpreted language*.
- Particularly useful for statistical analysis and data analytics.
- It is open source and has thousands of packages.
- Development Environments: Rstudio, Jupyter, console, etc.

- Let's do a Mini tutorial: Using R to create and plot a simple model in Rstudio

Scaling up your workflow

- Sometimes your laptop doesn't cut it. For example:
  - You need to process thousands of images.
  - You want to analyze millions of pages of text.
  - You want to do machine learning to train a robot.
  - You want to discover genes associated with a disease.
  - You want to run a climate model.
- …you're in luck! We have a FREE supercomputer at CU!
  - Porting code from your laptop isn't hard, and we can help.
  - See http://rc.colorado.edu/ for details

Resources for further learning (1)

- We offer courses!
  - Coding in R workshop series (Wednesdays 4p-6p., Feb 6-Apr 3)
  - Introduction to Python series (Tuesdays 9a-10:30a., Feb 26-Apr23)
  - Introduction to Bash (3p-5p, Monday Feb 11)
  - ….and more!

See https://www.colorado.edu/crdds/events & https://www.colorado.edu/rc/events

Resources for further learning (2)

- Personalized help:
  - Interdisciplinary Data Consult Hours (Tue 12p-2p, Norlin E206)
  - LISA Statistics Walk-in Hours (Tue & Thur 12p-2p, Norlin E206)
  - Supercomputing consultations (email rc-help@colorado.edu)
- Online software training: https://software-carpentry.org/lessons/

Questions?

Participant Feedback Survey:

http://tinyurl.com/curc-survey18

Slides and examples from this presentation:

https://github.com/ResearchComputing/coding_intro_spring_2019