

This is CS50x

OpenCourseWare

[David J. Malan \(https://cs.harvard.edu/malan/\)](https://cs.harvard.edu/malan/)

malan@harvard.edu

[f \(https://www.facebook.com/dmalan\)](https://www.facebook.com/dmalan)

[@ \(https://www.instagram.com/davidjmalan/\)](https://www.instagram.com/davidjmalan/)

[in \(https://www.linkedin.com/in/malan/\)](https://www.linkedin.com/in/malan/)

[Q \(https://www.quora.com/profile/David-J-Malan\)](https://www.quora.com/profile/David-J-Malan)

[🐦 \(https://twitter.com/davidjmalan\)](https://twitter.com/davidjmalan)

Lecture 9

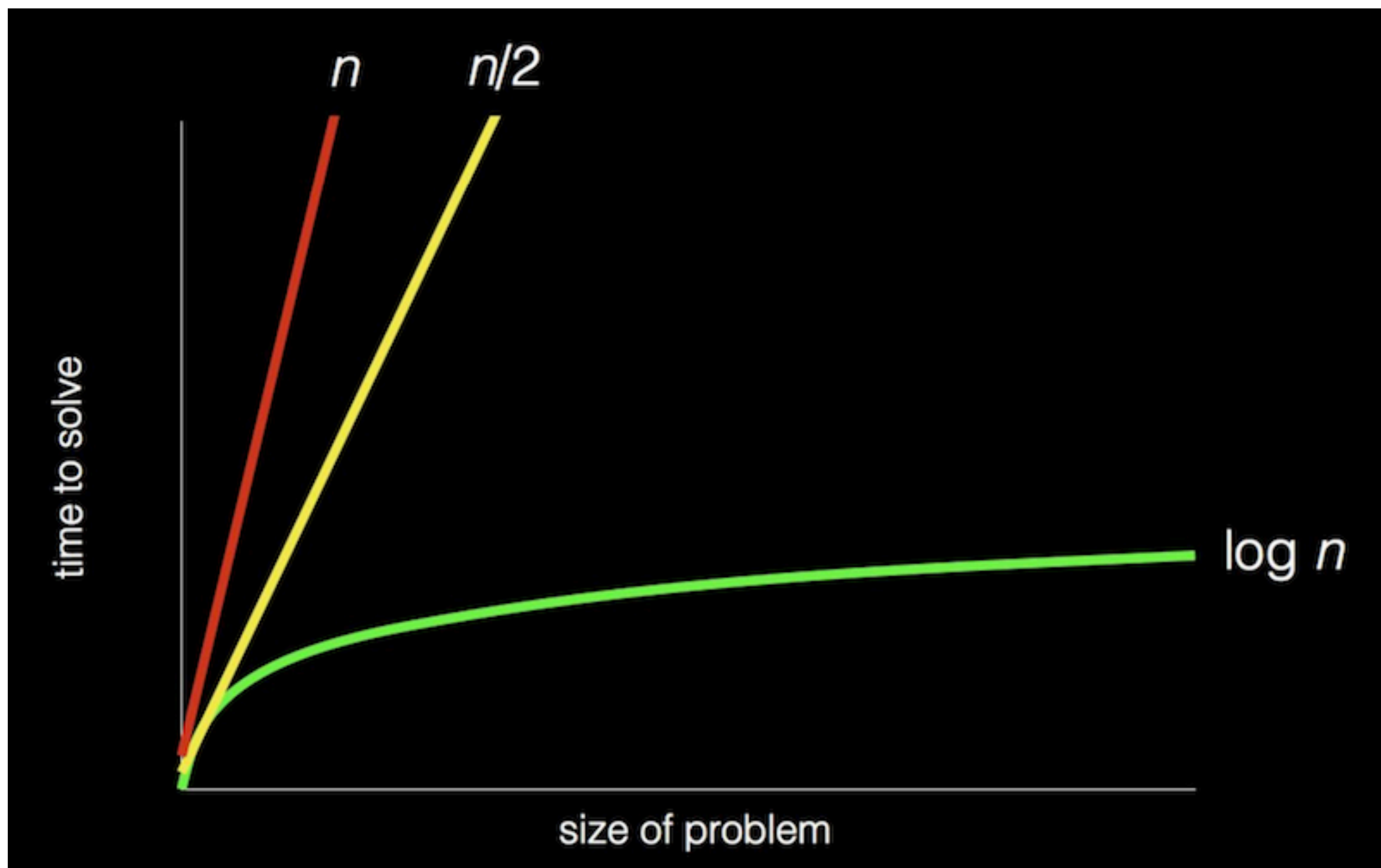
- [Last times](#)
- [Next times](#)
- [Jeopardy](#)

Last times

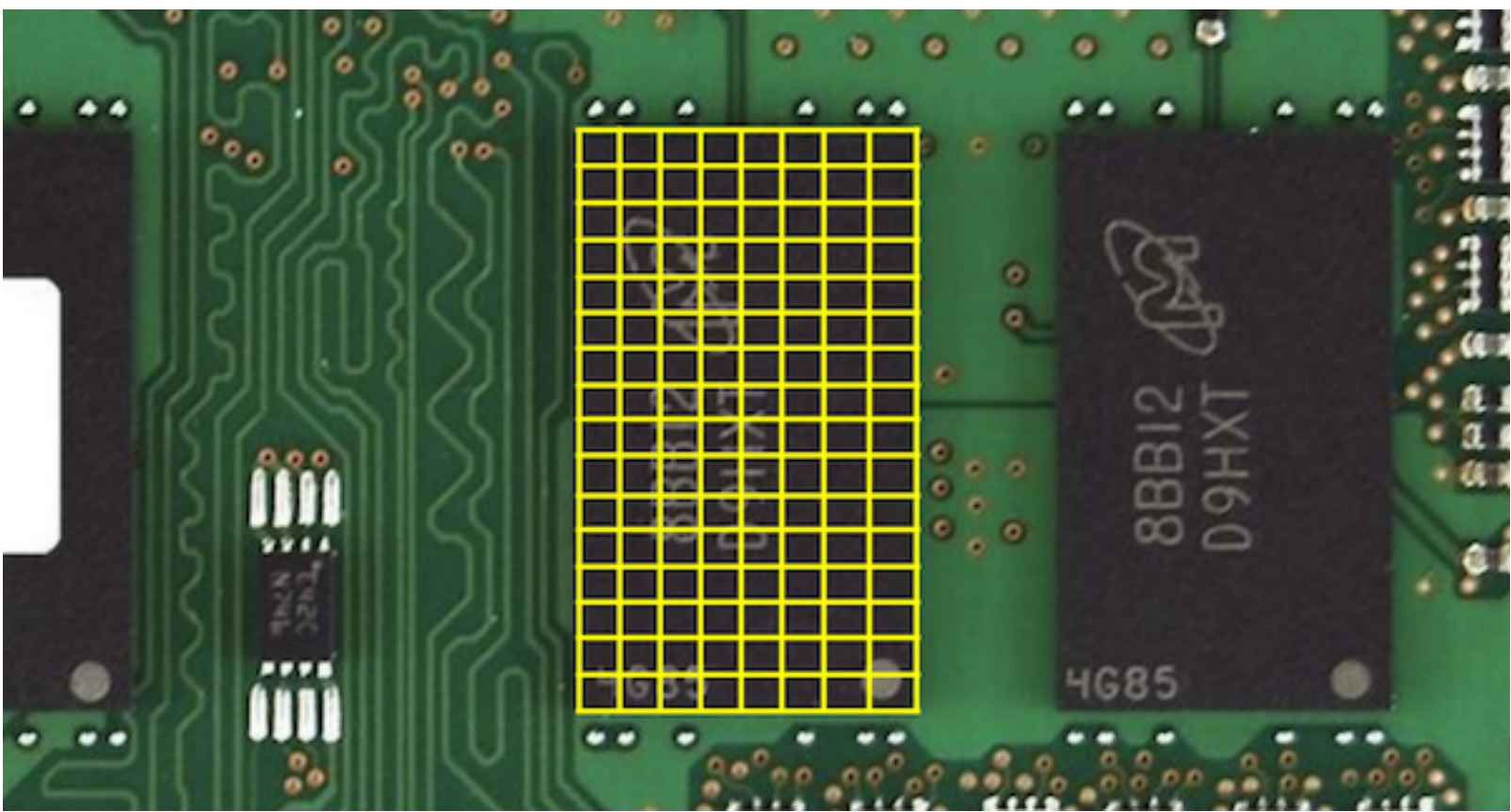
- In week 0, we said:
 - what ultimately matters in this course is not so much where you end up relative to your classmates but where you, in V, end up relative to yourself in Week 0
- And indeed, looking back now, we've learned new technologies and concepts every week. Even though we might not have e like we were completely comfortable, we've certainly gained experience and knowledge in more and more areas.
- In our first problem set, we probably found it difficult to print out a pyramid of # marks. But now, we're able to build an en application for CS50 Finance, with a back-end and a front-end, using a number of languages and frameworks.
- One of your classmates wrote, "Beside learning the new material, I felt that I started to be more 'comfortable with being uncomfortable' and being okay with working towards building a skill rather than just getting the right answer...", which bea encapsulates one of the goals of this course. Indeed, even in the real world, languages and technologies evolve over time, b come to appreciate principles of abstractions and tradeoffs and algorithms and data structures that will guide us in solving problems with computer science.
 - In fact, David himself had to learn Python a few years ago! And he too had to figure out how to build the Frosh IMs site ago, since web development wasn't part of the CS curriculum back then.
- A big thank you to all the staff, including the production team, the TFs and CAs, and course heads, without whom this course not be possible.
 - In particular: Cheng, who has been writing these notes; Veronica, who has been translating the course's materials in Sp and Brian, who has been an extraordinary head TF these past few years.
- In week 0, we learned that algorithms were a series of steps to solve a problem, given some inputs and outputs:

algorithms

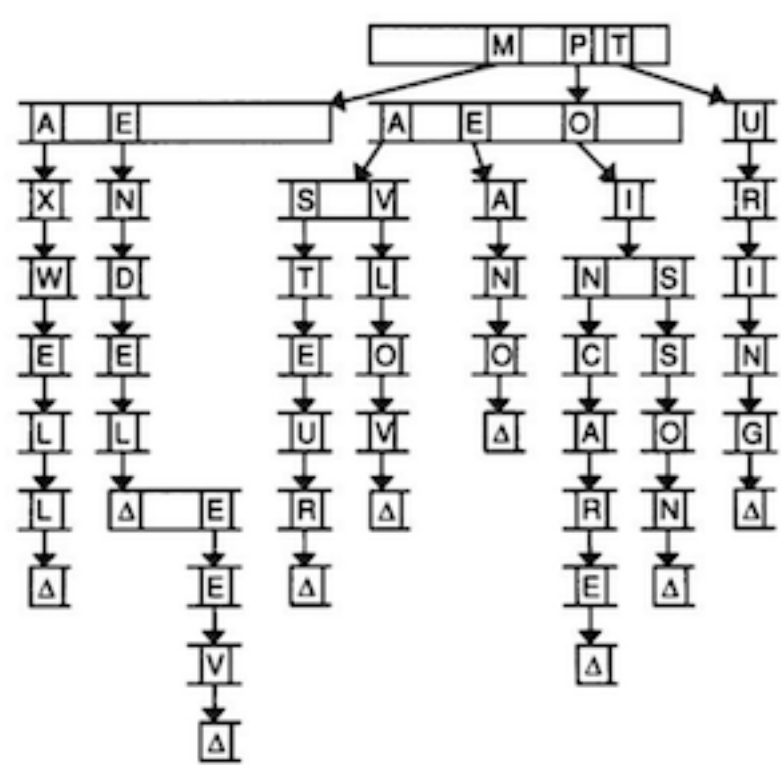
- And to represent those inputs and outputs in computers, we needed to understand bits, binary digits, atop which we did various abstractions like ASCII for text characters, Unicode for emoji, and RGB for pixels in images and videos.
- We wrote some algorithms down in pseudocode, and were already able to discuss running time of algorithms in terms of the number of steps required to solve a problem of size n :



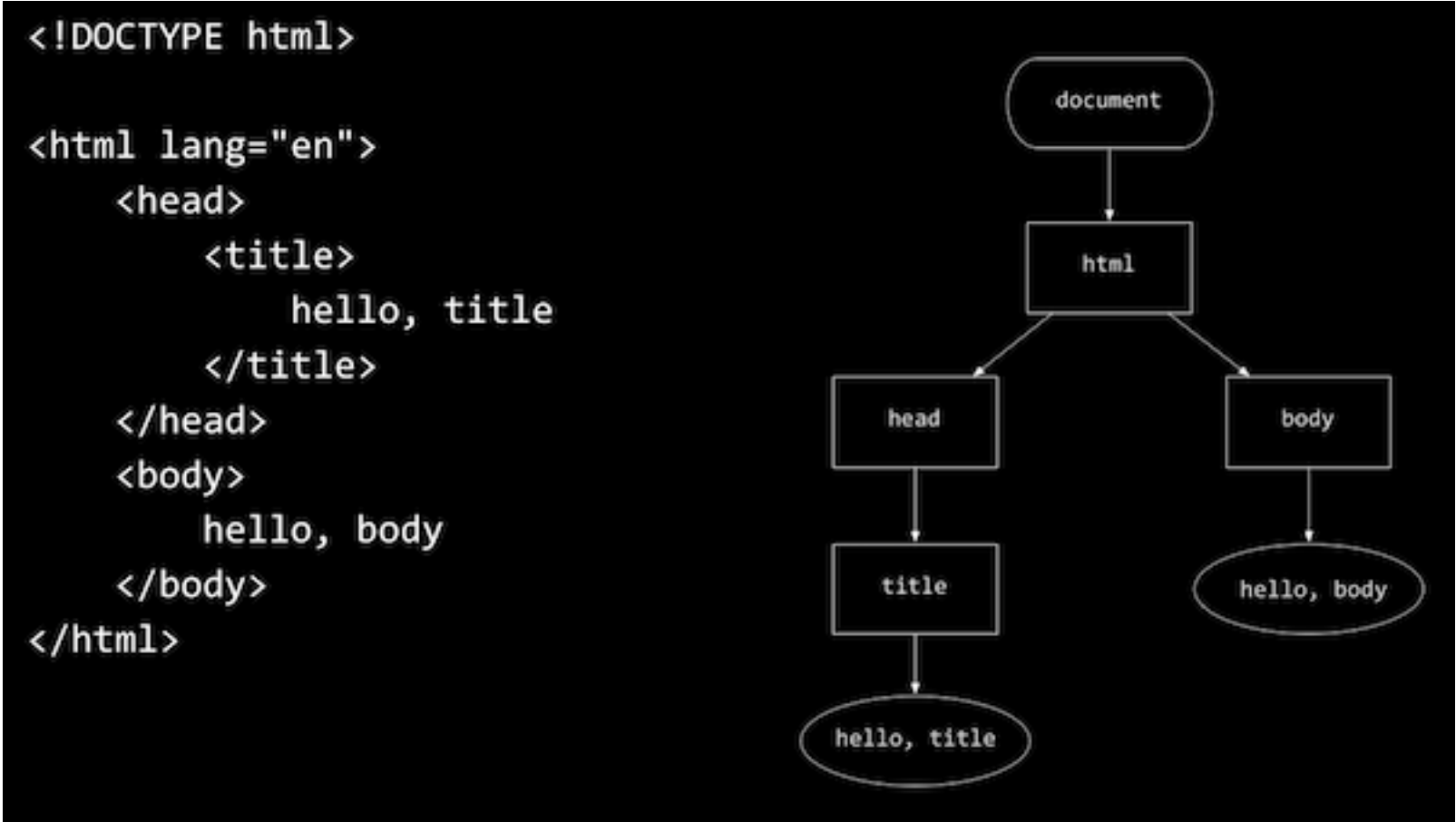
- Finally, we were introduced to Scratch, where we practiced using these new programming concepts.
- Then, in week 1, we transitioned to learning C, where we used only text to write source code, which was then compiled into machine code, binary that our computer could actually run.
- In week 2, we learned more programming features like functions, arguments, and discussed memory, where we could change the hardware of our computer, to store and work with data:



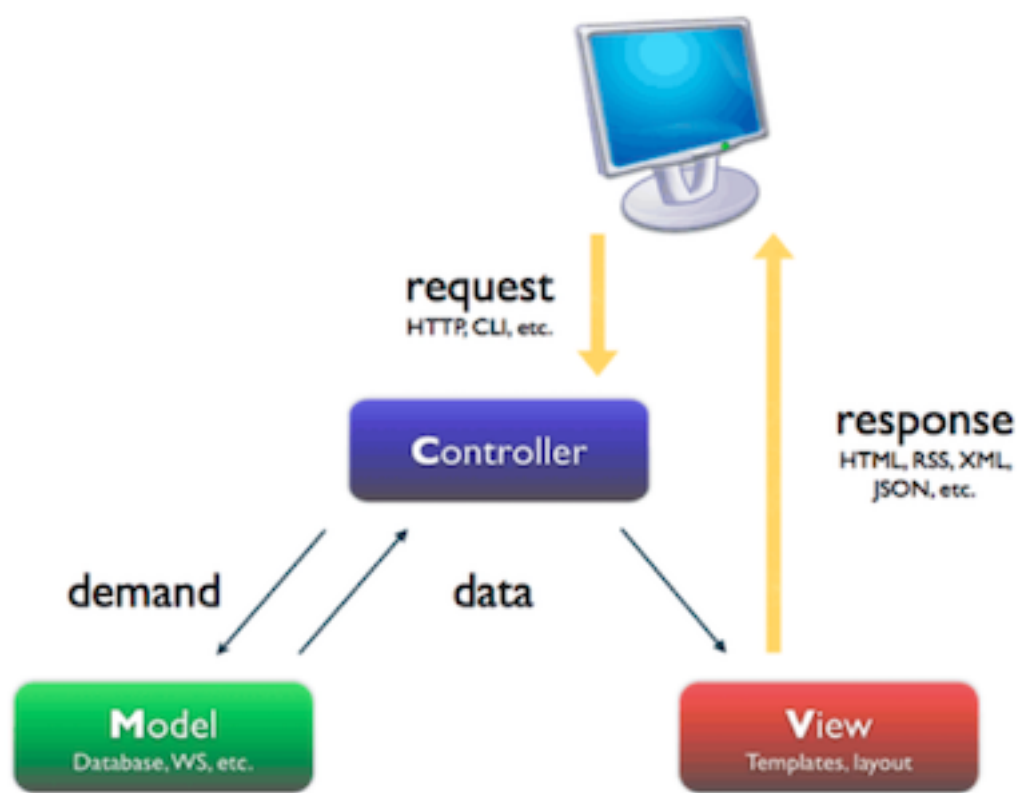
- In week 3, we learned about strings in C, which led us to discover pointers. In week 4, we used pointers to build data structures like a trie:



- In week 5, we learned about (many) web technologies like HTTP, DNS, IP, HTML, CSS, and JavaScript. We looked at request headers and response codes in HTTP, and we looked at HTML's Document Object Model (DOM) that represents a webpage in the browser:



- And as an aside, in one year's Harvard-Yale football game, some Yale students actually managed to prank much of the audience to hold signs that read "WE SUCK"!
- In week 6, we learned Python, a higher-level language that includes features like linked lists and hash tables, so we don't need to implement them ourselves before we start writing more useful programs.
- In week 7, we looked at one particular paradigm of web programming, Model-View-Controller, which is just one popular pattern for organizing our code:



- Finally, in week 8, we looked at SQL and databases more generally, with which we can query data efficiently and cleanly.

Next times

- In a few weeks, we're hosting the CS50 Hackathon, an evening at which students and staff will collaborate on final projects
- Then, we'll have the CS50 Fair, where we'll demo student projects for the entire campus to check out.
 - We watch a [short video \(https://www.youtube.com/watch?v=7NlaBl_jt2k\)](https://www.youtube.com/watch?v=7NlaBl_jt2k) of a Muppet submitting a final project just before the deadline!
- In the real world, we might not use the CS50 IDE, but command-line tools for macOS and Windows like:
 - [Xcode \(https://developer.apple.com/xcode/\)](https://developer.apple.com/xcode/)
 - [Windows Subsystem for Linux \(https://docs.microsoft.com/en-us/windows/wsl/about\)](https://docs.microsoft.com/en-us/windows/wsl/about)
- We should probably [learn about \(https://youtu.be/MJUJ4wbFm_A\)](https://youtu.be/MJUJ4wbFm_A) and use Git, a version-control software so we can track and manage changes to our source code, among other useful features.
- As students, we can sign up for [free private repositories \(https://education.github.com/discount_requests/new\)](https://education.github.com/discount_requests/new) on GitHub, a platform where we can store our code in the cloud.
- There are lots of text editors, too, like:
 - [Atom \(https://atom.io/\)](https://atom.io/)
 - [VS Code \(https://code.visualstudio.com/\)](https://code.visualstudio.com/)
 - [Vim \(http://valloric.github.io/YouCompleteMe/\)](http://valloric.github.io/YouCompleteMe/)
- We can host static websites with free services like:
 - [GitHub Pages \(https://pages.github.com/\)](https://pages.github.com/)
 - [Netlify \(https://www.netlify.com/\)](https://www.netlify.com/)
 - ...
- And web apps with services like:
 - [Heroku \(https://www.heroku.com/platform\)](https://www.heroku.com/platform)
 - [Wix Code \(https://www.wix.com/code/home\)](https://www.wix.com/code/home)
 - ...
- We can look into bigger providers, who have more sophisticated services, like:
 - [AWS \(https://aws.amazon.com/education/awseducate/\)](https://aws.amazon.com/education/awseducate/)
 - [Azure \(https://azure.microsoft.com/en-us/free/students/\)](https://azure.microsoft.com/en-us/free/students/)
 - [Google Cloud \(https://cloud.google.com/edu/\)](https://cloud.google.com/edu/)
- There are lots of sites where we can learn more about technology:
 - [https://www.reddit.com/r/programming \(https://www.reddit.com/r/programming\)](https://www.reddit.com/r/programming)
 - [https://news.ycombinator.com/ \(https://news.ycombinator.com/\)](https://news.ycombinator.com/)
 - [https://techcrunch.com/ \(https://techcrunch.com/\)](https://techcrunch.com/)

- <https://stackoverflow.com/> (<https://stackoverflow.com/>).
- <https://serverfault.com/> (<https://serverfault.com/>).
- <https://www.google.com/> (<https://www.google.com/>).
- ...
- CS50, too, has a presence online in various communities:
 - <https://www.reddit.com/r/cs50> (<https://www.reddit.com/r/cs50>).
 - <https://discord.gg/QYZOfZ6> (<https://discord.gg/QYZOfZ6>).
 - <https://www.facebook.com/groups/cs50> (<https://www.facebook.com/groups/cs50>).
 - <https://gitter.im/cs50/x> (<https://gitter.im/cs50/x>).
 - <https://www.instagram.com/cs50/> (<https://www.instagram.com/cs50/>).
 - <https://www.linkedin.com/groups/7437240/> (<https://www.linkedin.com/groups/7437240/>).
 - <https://www.quora.com/topic/Harvard-CS50> (<https://www.quora.com/topic/Harvard-CS50>).
 - <https://cs50x.slack.com/> (<https://cs50x.slack.com/>).
 - <https://www.snapchat.com/add/cs50> (<https://www.snapchat.com/add/cs50>).
 - <http://cs50.stackexchange.com/> (<http://cs50.stackexchange.com/>).
 - <https://twitter.com/cs50> (<https://twitter.com/cs50>).
- In fact, after this class, we are now qualified to be staff for CS50, since we'll (hopefully!) take more CS courses in the spring become even more knowledgeable by next fall.

Jeopardy

- We take a few staff and student volunteers, and play Jeopardy! These questions and answers were submitted by students as suggested quiz questions.
- The first question: What institution developed Scratch?
 - Answer: MIT
- Question: What color is the cat?
 - Answer: Orange
- Question: What is running time of merge sort?
 - Answer: $O(n \log n)$
- Question: What is the tool you can run if your program has memory leaks?
 - Answer: Valgrind
- Question: What is rubber duck debugging?
 - Answer: When you read your code to a rubber duck and try to find the mistake.
- Question: Is the C the absolute worst?
 - Answer: Yes
- Question: What is the name of an animal you don't want to find under your pillow? (That shares a name with a programming language.)
 - Answer: Python
- Question: Who is your friend?
 - Answer: Bootstrap
- Question: What is the (HTTP response) code for OK?
 - Answer: 200
- Question: What do you call the situation where a structure refers to itself in its definition?
 - Answer: Recursion
- Question: In what state, besides Massachusetts, does Google sometimes think we reside?
 - Answer: Kansas
- Thanks for joining us!

