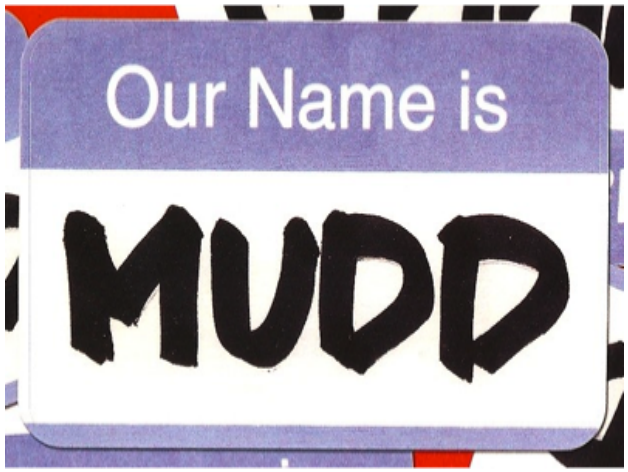
Introduction

Day 1, Session 1

Welcome!

MyCS: *Middle-years' Computer Science*

July 9- July 13, 2012



- Housekeeping: where? when?
- Introductions: who?
- Welcome: what? (1) CS, (2) ECS, and (3) MyCS

- Account information, forms, binders, etc.

- Dive in!

Housekeeping and Schedule

Restrooms? ~ SW corner

Schedule

	Monday	Tuesday	Wednesday	Thursday	Friday
Session 1: 9am-10:30am	Introduction: Computers?! Finding facts	Introduction to computational problem solving	Algorithms: searching and sorting them out	Using the web: social responsibility	Wrap-up and looking forward
Session 2: 10:45am-noonish	Introduction to programming: Making music with Scratch	Scratch: Position (and more debugging!)	Scratch: Ifs and Elses	Creating the web: HTML and CSS	field trip
Lunch	noon to 1pm				
Session 3: 1pm-2:15pm	Computers: Getting the inside view	Bits and binary... and Lego	Limits of computing: AI, 20 questions, and life	Security and passwords	field trip
Session 4: 2:30pm-4pm	Scratch: Movement and debugging	Scratch: Images	Scratch: Broadcasting	Javascript and final recipes	field trip

Introductions? ~ coming next...

Introductions

- Name tags...
- Our own introductions...
- Saying hello to our pairs around the room...
- Please take the "Welcome!" survey

@ Google docs: login **`2012mycs`** passwd **`csisfun!`**

from the 90's:

Mike Erlinger

Zach Dodds

The team!

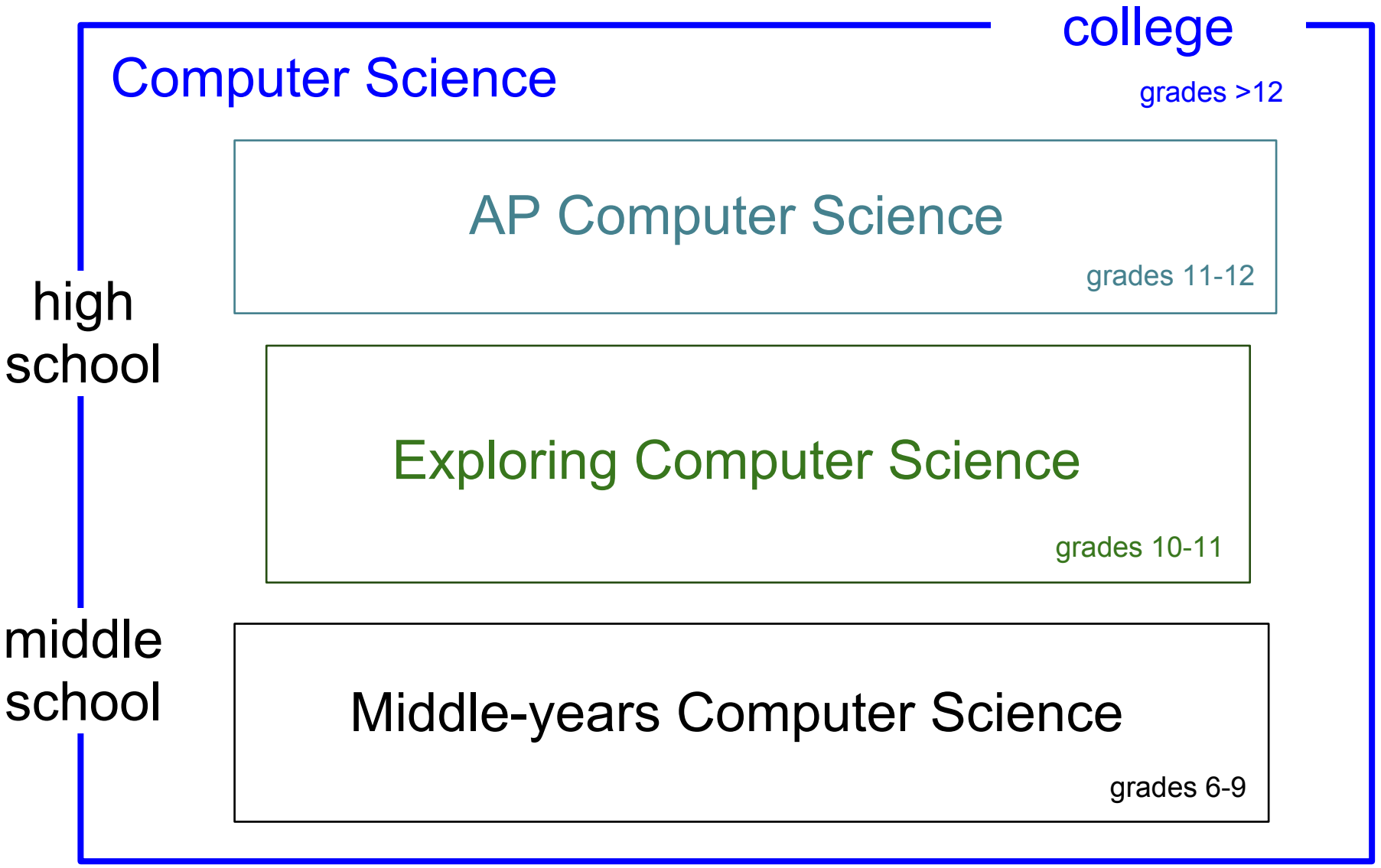
CS, ECS, and MyCS

Middle-years
Computer Science

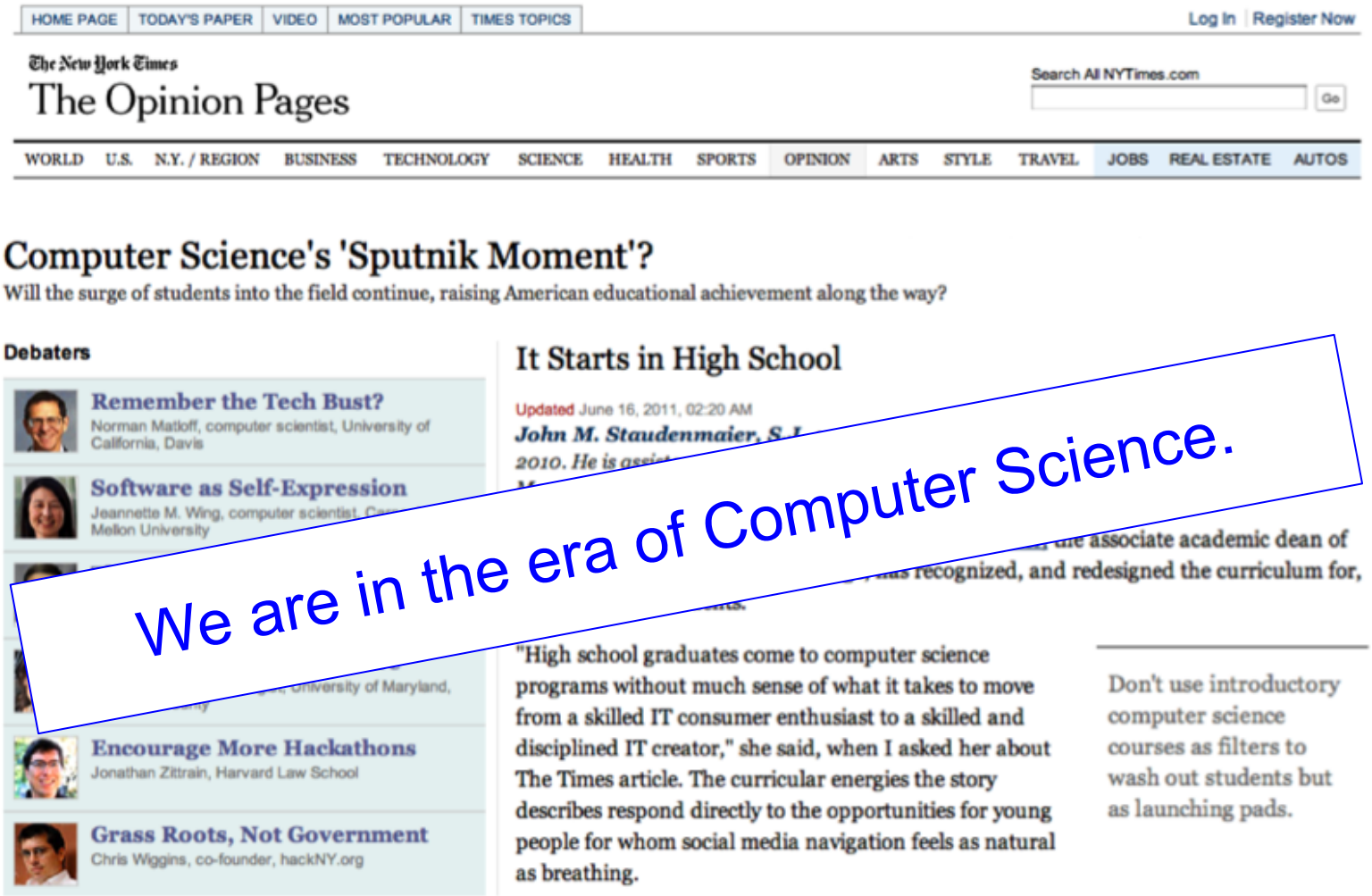
Exploring Computer Science

Computer Science

CS, ECS, and MyCS



CS, ECS, and MyCS



We are in the era of Computer Science.

http://www.nytimes.com/roomfordebate/2011/06/15/computer-sciences-sputnik-moment/it-starts-in-high-school

CS, ECS, and MyCS

Table 1. Occupations with the fastest growth

Occupations	Percent change	Number of new jobs *(in thousands)*	Wages (May 2008 median)	Education/training category
Biomedical engineers	72	11.6	$ 77,400	Bachelor's degree
Network systems and data communications analysts	53	155.8	71,100	Bachelor's degree
Home health aides	50	460.9	20,460	Short-term on-the-job training
Personal and home care aides	46	375.8	19,180	Short-term on-the-job training
Financial examiners	41	11.1	70,930	Bachelor's degree
Medical scientists, except epidemiologists	40	44.2	72,590	Doctoral degree
Physician assistants	39	29.2	81,230	Master's degree
Skin care specialists	38	14.7	28,730	Postsecondary vocational award
Biochemists and biophysicists	37	8.7	82,840	Doctoral degree
Athletic trainers	37	6.0	39,640	Bachelor's degree
Physical therapist aides	36	16.7	23,760	Short-term on-the-job training
Dental hygienists	36	62.9	66,570	Associate degree
Veterinary technologists and technicians	36	28.5	28,900	Associate degree
Dental assistants	36	105.6	32,380	Moderate-term on-the-job training
Computer software engineers, applications	34	175.1	85,430	Bachelor's degree
Medical assistants	34	163.9	28,300	Moderate-term on-the-job training
Physical therapist assistants	33	21.2	46,140	Associate degree
Veterinarians	33	19.7	79,050	First professional degree

Who reads these charts anyway?

Two of the fastest growing detailed occupations are in the computer specialist occupational group. Network systems and data communications analysts are projected to be the second-fastest-growing occupation in the economy. Demand for these workers will increase as organizations continue to upgrade their information technology capacity and incorporate the newest technologies. The growing reliance on wireless networks will result in a need for more network systems and data communications analysts as well. Computer applications software engineers also are expected to grow rapidly from 2008 to 2018. Expanding Internet technologies have spurred demand for these workers, who c

http://www.bls.gov/oco/ocos303.htm

CS, ECS, and MyCS

CS is exciting - and it needs to be!

TBD: Who will **consume** vs. **create** it?

http://www.nytimes.com/2011/06/11/technology/11computing.html?scp=2&sq=computer%20science&st=cse

January, 2012 at the Victoria Gardens Apple Store

CS, ECS, and MyCS

Choose a role: *consumption* vs. *contribution*

it's not really *vs.*

January, 2012 at the Victoria Gardens Apple Store

CS, ECS, and MyCS

Version 4.0

Course Overview

Goals

from page 5...

Exploring Computer Science is designed to introduce students to the breadth of the field of computer science through an exploration of engaging and accessible topics. Rather than focusing the entire course on learning particular software tools or programming languages, the course is designed to focus the conceptual ideas of computing and help students understand why certain tools or languages might be utilized to solve particular problems. The goal of *Exploring Computer Science* is to develop in students the computational thinking practices of algorithm development, problem solving and programming within the context of problems that are relevant to the lives of today's students. Students will also be introduced to topics such as interface design, limits of co

*ECS's Goal: to **democratize** CS*

Th Los An .4% Pacific Islander, 2.3% Filipino, 73.0% Latino, 10.9% African American, 8.8% White, and .6% Other or multiple responses. Over 38% of students are English-language learners, with most English language learners' students speaking Spanish as their primary language. Furthermore, 74% of students qualify for free or reduced lunches.

Everyone should have a chance to help *author* and *understand* the computation we use everyday.

CS, ECS, and MyCS

LA Unified students taking ECS:

2009	306
2010	922
2011	1377
2012	2000+

Other cities starting, as well...

We are working to bring this to the I.E. --
and PUSD in particular, *only a bit earlier*

CS, ECS, and MyCS

MyCS is an ECS-inspired course developed for middle-school and "middle-years" audiences.

what

- engaging CS ideas and activities
- hands-on skills, emphasizing ***creativity***
- can be ***spliced*** with technology courses, typing courses, and other existing electives

---

how

- ECS curriculum with emphasis on ***skills***
- we are looking for what will work in ***one term***
- *we'd love your help* creating a ***usable*** MyCS!

Exploring Computer Science

ECS

4-5 week units

Non-programming

What is CS? ~ computers and computing

Data and algorithms ~ the heart of CS

2 modules of problem-solving

Programming

The web ~ web pages, and web applications

Scratch ~ programming with sound and graphics

2 modules of programming

18 weeks total

Exploring Computer Science

ECS

4-5 week units

Non-programming

What is CS? ~ computers and computing

Data and algorithms ~ the heart of CS

Programming

The web ~ web pages, and web applications

Scratch ~ programming with sound and graphics

Is this "real"?

2 modules of problem-solving

too abstract...

Aargh!

2 modules of programming

smoother Scratch?

Scratch > Web

18 weeks total

Two 18-week schedules

Exploring Computer Science

ECS

4-5 week units

4-7 days spiraling

MyCS

Middle-years Computer Science

Non-programming

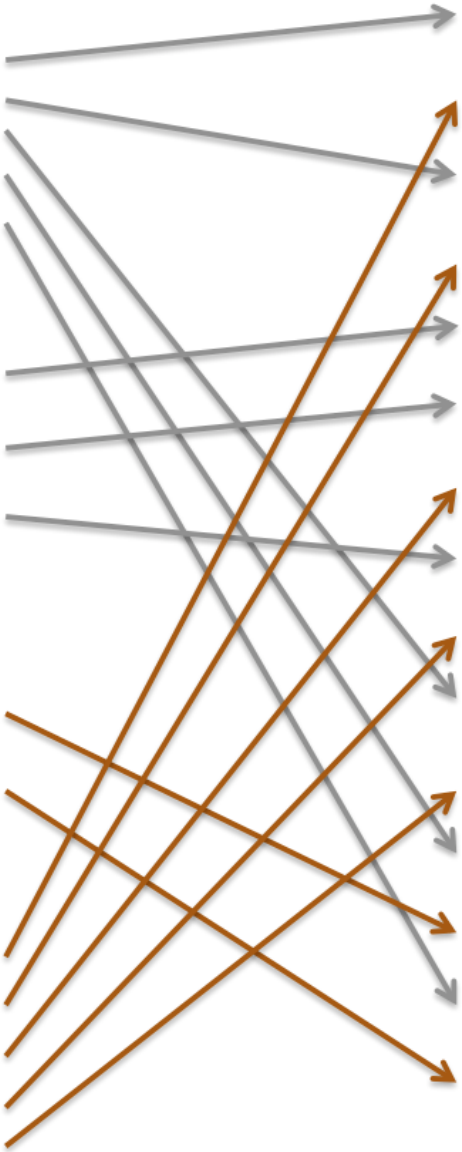
What is CS? ~ computers and computing

Data and algorithms ~ the heart of CS

Programming

The web ~ web pages, and web applications

Scratch ~ programming with sound and graphics



CS: what and why?

Scratch ~ sound

What is a computer?

Scratch ~ movement

CS as problem-solving

Encoding data

Scratch ~ coordinates

Algorithms ?!

Scratch ~ if/else

Limits of computers

Scratch ~ projects

Using the web ***well***

Web: HTML/CSS

Web safety/security

Web: Javascript

Games and extras

4-7 days spiraling

MyCS

Middle-years Computer Science

~15 modules interleaving problem-solving and programming

18 weeks total

- CS: what and why?
- **Scratch** ~ sound
- What is a computer?
- **Scratch** ~ movement
- CS as problem-solving
- Encoding data
- **Scratch** ~ coordinates
- *Algorithms* ?!
- **Scratch** ~ if/else
- Limits of computers
- **Scratch** ~ projects
- Using the web ***well***
- **Web: HTML/CSS**
- Web safety/security
- **Web: Javascript**
- Games and extras

Two 18-week schedules

Exploring Computer Science

ECS

4-5 week units

Non-programming

What is CS? ~ computers and computing

Data and algorithms ~ the heart of CS

Programming

The web ~ web pages, and web applications

Scratch ~ programming with sound and graphics

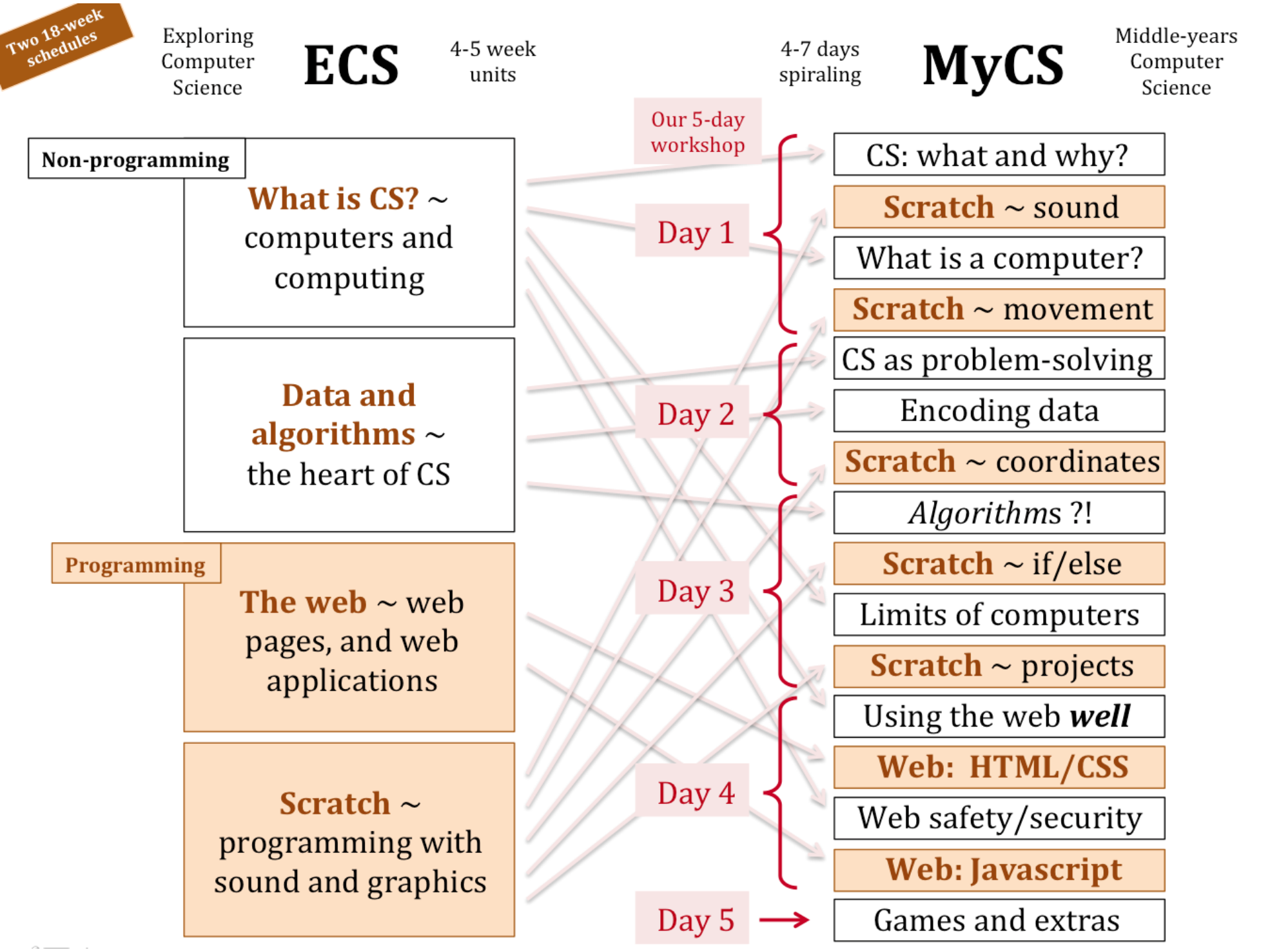
4-7 days spiraling

MyCS

Middle-years Computer Science

Our 5-day workshop

Day 1	CS: what and why?
	Scratch ~ sound
	What is a computer?
	Scratch ~ movement
Day 2	CS as problem-solving
	Encoding data
	Scratch ~ coordinates
Day 3	*Algorithms* ?!
	Scratch ~ if/else
	Limits of computers
	Scratch ~ projects
Day 4	Using the web ***well***
	Web: HTML/CSS
	Web safety/security
	Web: Javascript
Day 5 →	Games and extras


Two 18-week schedules
Exploring Computer Science
ECS
4-5 week units
4-7 days spiraling
MyCS
Middle-years Computer Science
Non-programming
What is CS? ~ computers and computing
Data and algorithms ~ the heart of CS
Programming
The web ~ web pages, and web applications
Scratch ~ programming with sound and graphics
Our 5-day workshop
Day 1
Day 2
Day 3
Day 4
Day 5
CS: what and why?
Scratch ~ sound
What is a computer?
Scratch ~ movement
CS as problem-solving
Encoding data
Scratch ~ coordinates
Algorithms ?!
Scratch ~ if/else
Limits of computers
Scratch ~ projects
Using the web well
Web: HTML/CSS
Web safety/security
Web: Javascript
Games and extras

Natural place to start:

What is computing?

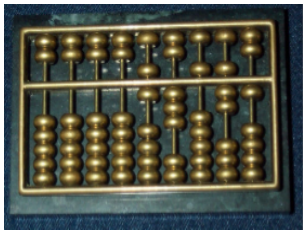
What is a computer?

(1) For each of these items, decide if it's a computer or not.

ACTIVITY

☐ Computer

☐ Not a computer



☐ Computer

☐ Not a computer

☐ Computer

☐ Not a computer

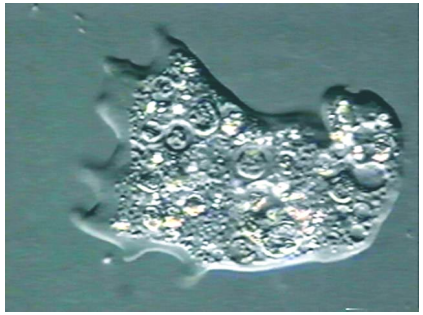
☐ Computer

☐ Not a computer

☐ Computer

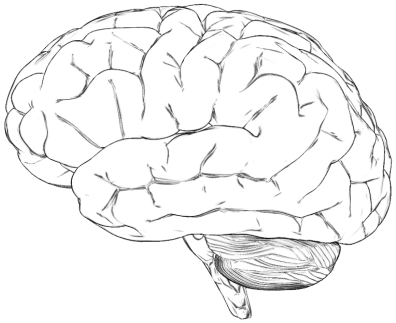
☐ Not a computer

ACTIVITY



☐ Computer

☐ Not a computer



☐ Computer

☐ Not a computer

☐ Computer

☐ Not a computer

☐ Computer

☐ Not a computer

Computer? Or, not a computer?

What is a computer / computing?

First, think of two other examples of computers - be creative!

1.
2.

Also, think of two other examples of *non*-computers

1.
2.

Now, list three traits or characteristics that make something a "computer":

1.

2.

3.

And -- list three traits that make something ***not*** a computer!

1.

2.

3.

could this be made to work in a class?

CS and life?

Are there any connections there?

CS and life...

... and our sponsor!

Web Search

certainly sets the standard for
"information interactions"

especially freeing in education

Giving fish vs. teaching fishing!

Web *Skills*

What are some of the strategies *you* use to make web-searching fast & effective?

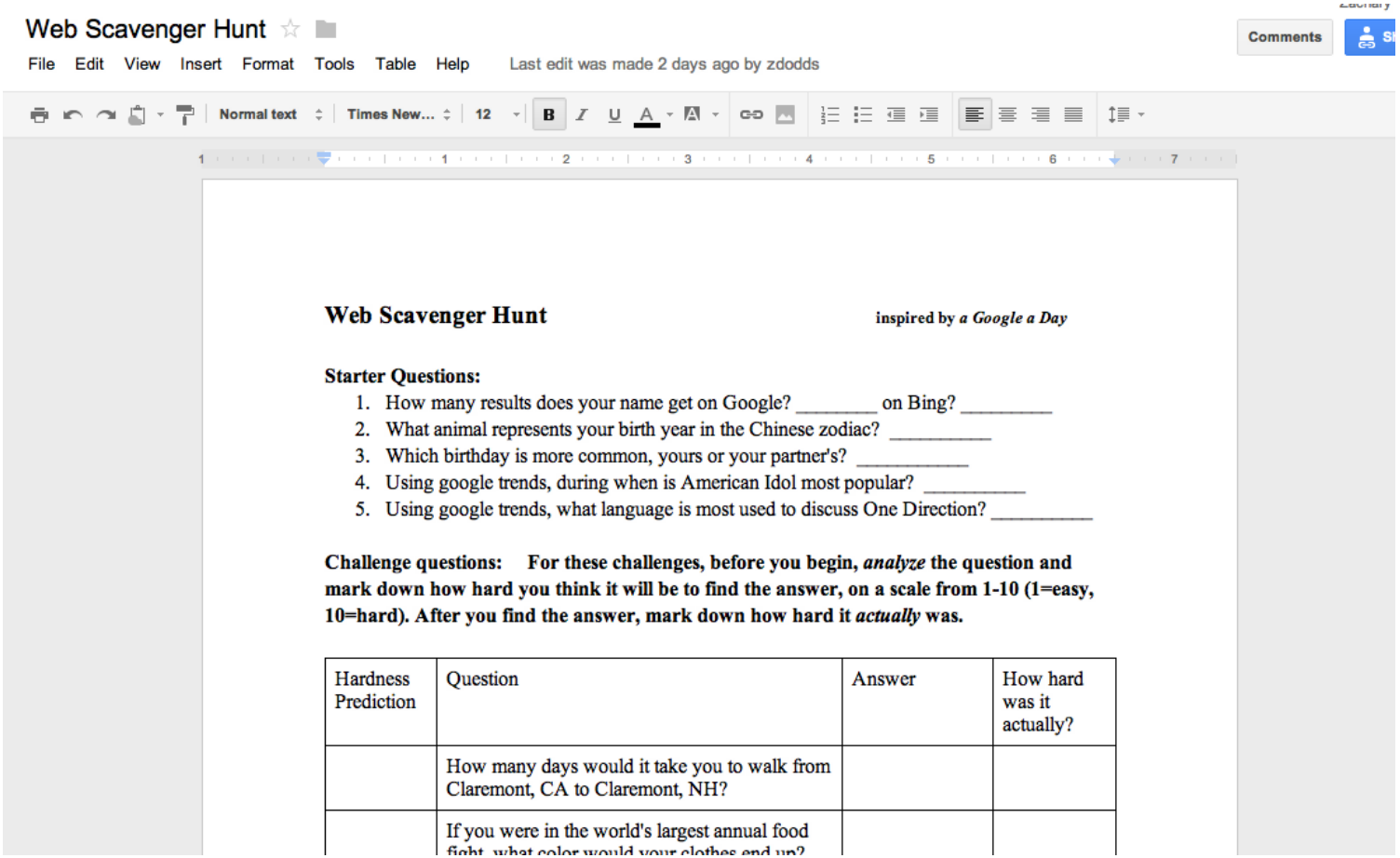
let's return to this question -- after you try things out!

Web Scavenger Hunt

With your partner, find answers to these challenge questions...

For the later ones, estimate how difficult you think the question is (both before & after)

Web Scavenger Hunt



Web Scavenger Hunt

File Edit View Insert Format Tools Table Help Last edit was made 2 days ago by zdodds

Comments

Normal text | Times New... | 12

Web Scavenger Hunt inspired by *a Google a Day*

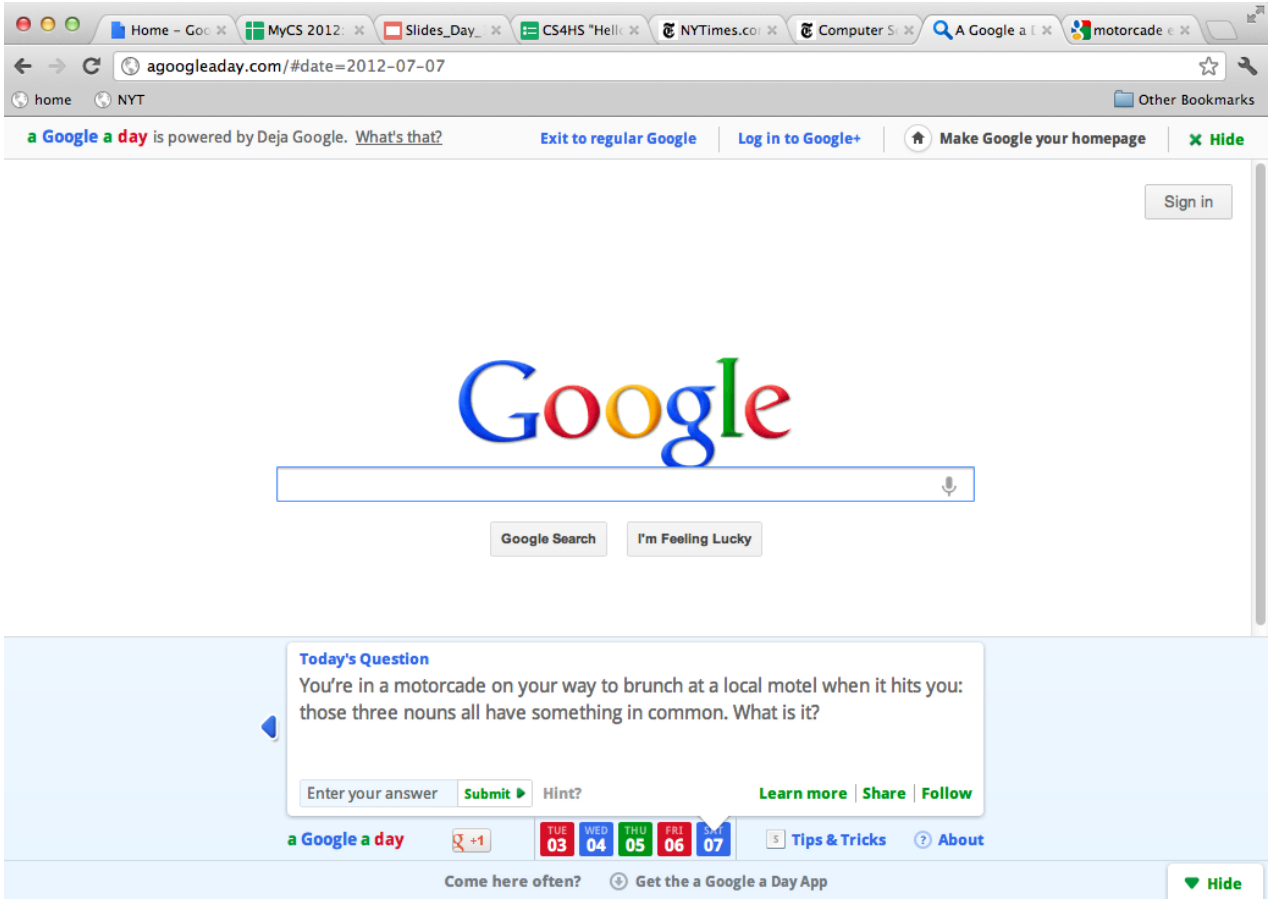
Starter Questions:

1. How many results does your name get on Google? _____ on Bing? _____
2. What animal represents your birth year in the Chinese zodiac? _____
3. Which birthday is more common, yours or your partner's? _____
4. Using google trends, during when is American Idol most popular? _____
5. Using google trends, what language is most used to discuss One Direction? _____

Challenge questions: For these challenges, before you begin, *analyze* the question and mark down how hard you think it will be to find the answer, on a scale from 1-10 (1=easy, 10=hard). After you find the answer, mark down how hard it *actually* was.

Hardness Prediction	Question	Answer	How hard was it actually?
	How many days would it take you to walk from Claremont, CA to Claremont, NH?		
	If you were in the world's largest annual food fight, what color would your clothes end up?		

Google-a-Day



Sample Scavenger Hunt

In your group, use the internet to find the following items. For each item i

1. A picture of the mayor of your town or city
2. A bus schedule
3. The address of the Chamber of Commerce for your town or city
4. A map of your state—and you have to point out where your town
5. A copy of the front page of your town's or city's web site
6. Something in writing that tells how many people live in the city
7. A picture of any historical landmark in the city
8. A picture of your congressman
9. A program or flyer from a local arts event
10. The names of all the city council members
11. Something that gives information about your local hospital
12. A list of schools in your town or city
13. The phone number of the local police department
14. Anything with the colors or mascot of a local college or communit
15. A picture of the state flag
16. A picture of the state bird
17. A schedule of activities or a pamphlet from a local nursing home o
18. A sticker or button from a local election
19. A list of safety tips from the local fire department
20. A speech by your governor

ECS's web challenge

Web Skills

What are some of the strategies you use to make web-searching fast & effective?

are your students already good at this -- or is this something worth reinforcing?

CS + life: *other connections*?

What else do your students already do that is determined or dominated by computer science?

Intro to Programming

Scratch: Music

Day 1, Session 2

Computer Programming

- Programming is the *stereotypical* CS activity.
- But CS is really not about screenfuls of crazy text...

```
PLEASE DO ,1 <- #13
DO ,1 SUB #1 <- #238
DO ,1 SUB #2 <- #112
DO ,1 SUB #3 <- #112
DO ,1 SUB #4 <- #0
DO ,1 SUB #5 <- #64
DO ,1 SUB #6 <- #238
DO ,1 SUB #7 <- #26
DO ,1 SUB #8 <- #248
DO ,1 SUB #9 <- #168
DO ,1 SUB #10 <- #24
DO ,1 SUB #11 <- #16
DO ,1 SUB #12 <- #158
DO ,1 SUB #13 <- #52
PLEASE READ OUT ,1
PLEASE GIVE UP
```

Intercal

```
                    v
>v"Hello world!"0<
,:
^_25*,@
```

Befunge

What ***popular*** programming languages have you used/heard of?

Programming languages

- What's popular now? (tiobe.com)

Position Jun 2012	Position Jun 2011	Delta in Position	Programming Language	Ratings Jun 2012	Delta Jun 2011	Status
1	2	↑	C	17.725%	+1.45%	A
2	1	↓	Java	16.265%	-2.32%	A
3	3	=	C++	9.358%	-0.47%	A
4	7	↑↑↑	Objective-C	9.094%	+4.66%	A
5	4	↓	C#	7.026%	+0.18%	A
6	6	=	(Visual) Basic	6.047%	+1.32%	A
7	5	↓↓	PHP	5.287%	-1.31%	A
8	8	=	Python	3.848%	-0.05%	A
9	9	=	Perl	2.221%	-0.09%	A
10	12	↑↑	Ruby	1.683%	+0.20%	A
11	11	=	JavaScript	1.474%	-0.03%	A

Intro to Computer Programming

Here are two more *"Hello, World!"* programs:

```
print 'Hello, World!'
```

Python

```
int main()
{
  std::cout << "Hello, world!";
}
```

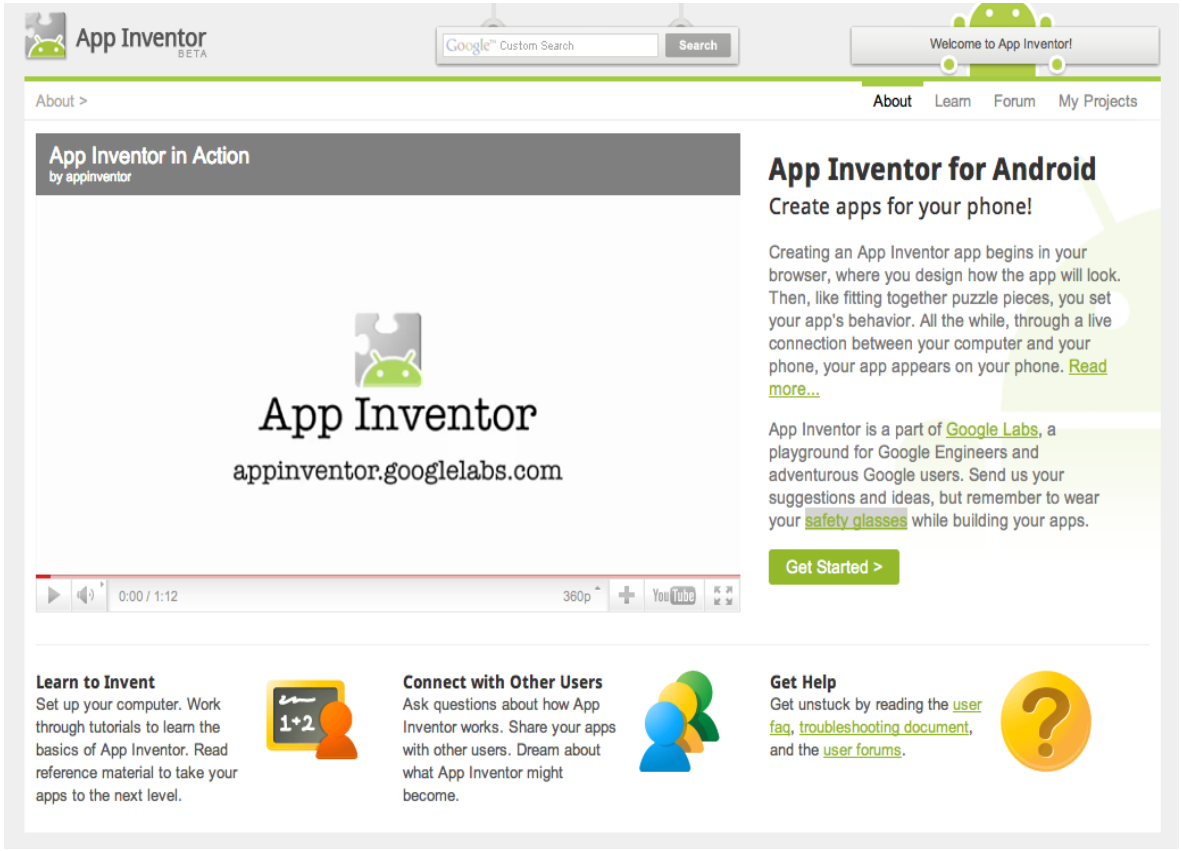
C++

CS is fundamentally about *problem-solving*...
... and programming is just its ***language***

Programming is <u>much</u> more like learning a new ***human*** language than many suspect: it's just ***translation***, *but easier!*

Programming languages

- What does the ***future*** of programming look like?



Graphical interfaces are ever more popular in education ***and industry***.

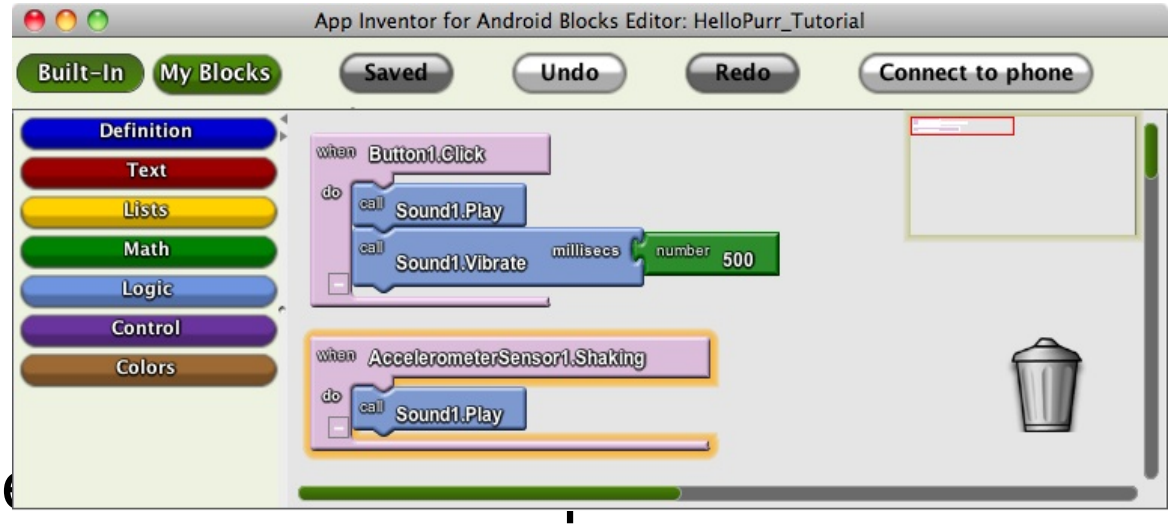
Programming languages

- A "Hello, Kitty" Android Program:

What are these bricks saying?

Programming ***ideas***

- ***Actions*** are the basic building blocks (literally!)...



- The
- The actions can be simple.
- The combinations can be powerful!

Scratch!

- Developed at MIT
- Helps anyone learn to program without worrying about syntax
- Its ***graphical interface*** includes a set of blocks that snap together

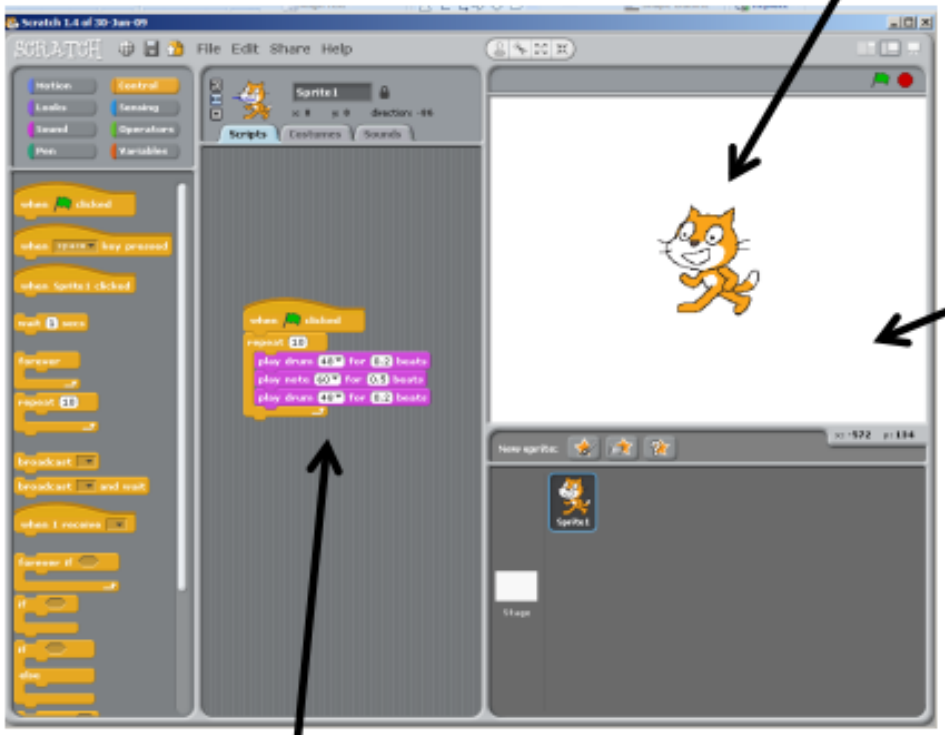
Lingo

A picture of a character or object that can be controlled.

Sprite

Block tabs

Click on these to get different sets of blocks.



Stage

The place where the sprites performs your scripts.

Blocks

Individual instructions for the sprite. Can be combined to make scripts.

Script

A set of instructions for a sprite to follow.

Notes!

The play note block plays a **musical** note. It has 2 **variables**:

1. You can change the **note**.
2. You can change how **long** the note plays (in beats).

Drums!

The play note block plays a **percussion** note. It has 2 variables:

1. You can change the **instrument**.
2. You can change how **long** the note plays (in beats).

Rests!

The rest block makes the script **pause**. It has 1 variable:

1. You can change how long the pause **lasts** (in beats).

Make a song!

Use notes, drums, and rests to create a short song.

When you are done, make sure to save your program -- *saving to the desktop is OK*

Then, have the other partner make a second song -- and play both songs for a nearby group!

Repeats!

The repeat block tells the computer to do the same thing **over and over** again. It has 1 variable:

1. You can change how many times to repeat the **blocks** inside the repeat block.

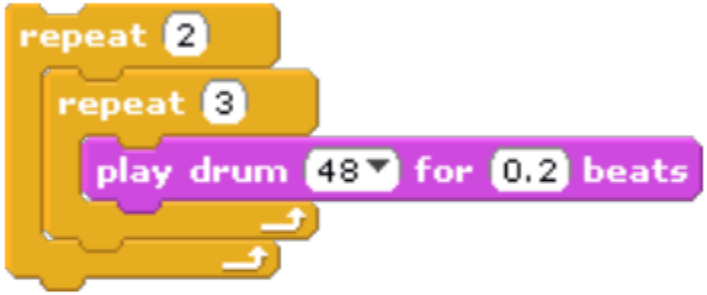
Repeats, continued!

Other blocks can be placed **inside of** the repeat block. Everything inside the repeat block will be repeated as many times as you want.

Nested Repeats!

A nested repeat is a repeat block placed **inside** another repeat block.

Nested Repeats, continued!



Nested repeats **multiply** together.

1. This drum note will play **6** times.

Tempo!

The tempo block increases the **speed** at which notes play. It has 1 variable:

1. You can change how many beats play **per minute**.

Review: Set Tempo!

The tempo block sets the speed at which notes play. It has 1 variable:

1. You can change how many beats play per minute.

When using the set tempo block, it **does not matter** what the current tempo is. Set tempo will set the new tempo to the number you **type in**.

Change Tempo!

The change tempo block **increases** **or** **decreases** the tempo by whatever amount you want. It has 1 variable:

1. You can change how much to increase or decrease the tempo. You may use **positive** or **negative** numbers.

When using the change tempo block, it **does matter** what the current tempo is. Set tempo will **add or subtract** from the current tempo.

Volume works the same way!

1. Use the **checkbox** next to the “volume” block to always see the volume.

2. Use the **set volume** block to reset the volume to the level you want.

3. Use the **change volume** block to increase or decrease the volume by a certain amount.

Song Improvements

Use repeats, tempo, and volume to make improvements to your songs or create a new one...

... either way, explore all of the tools!

"Mapping" Commands to the Keyboard

The "when key pressed" block is a **control** block. It **runs** a script when a button on the **keyboard** is pressed. It has 1 variable:

1.You can change **which key** will start the script.

The "when key pressed" block only works at the **beginning** of a script. (You can think of it as a "hat" for the script.)

Mapping Commands to the Keyboard!

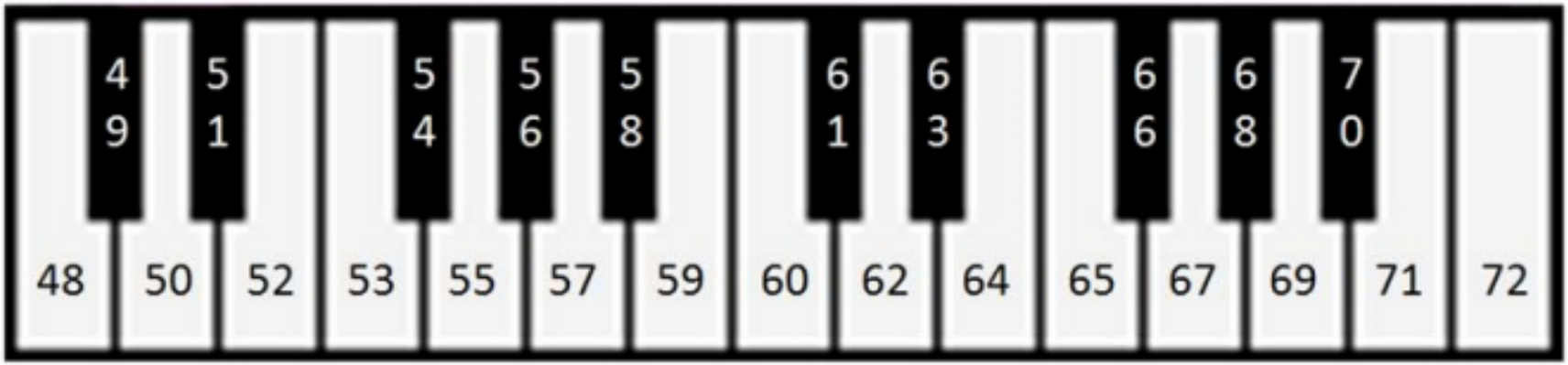
To control a script with the keyboard, place a "when key pressed" block on **top** of it and **choose** which key will run the script.

When a key on the keyboard starts a script, we say that script has been "**mapped**" to that key.

Keyboard Challenge

In a new Scratch project, create a ***keyboard***.

That is, make each key in a row of keys on your computer keyboard correspond to a note in a scale. *Want more? Try different instruments!*



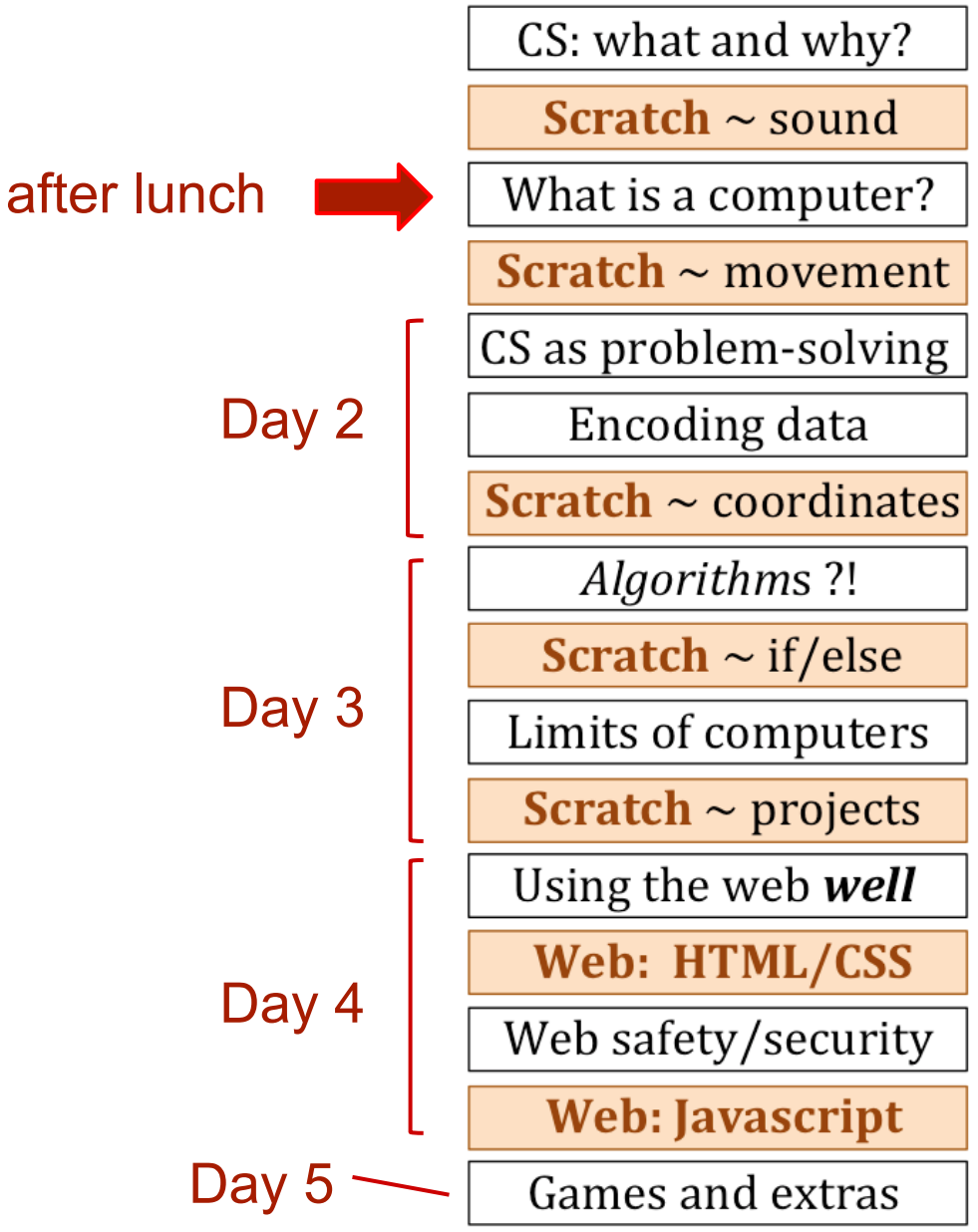
The MyCS curriculum starts with music -- would that work? How could it be sold more effectively?

Where we are...

4-7 days spiraling

MyCS

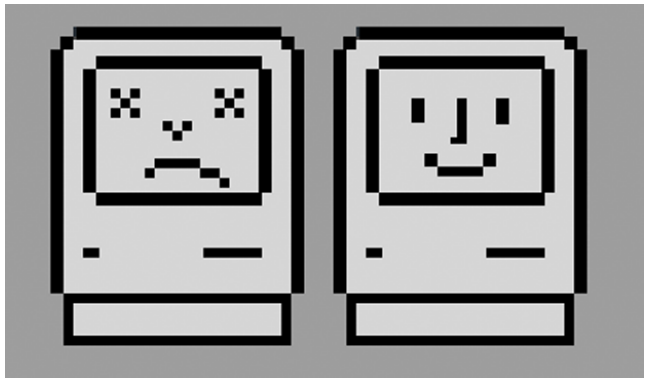
Middle-years Computer Science



Computer Disassembly

Day 1, Session 3

Behind the "curtain"...



Friend, foe, *or fork* ?

Computers can seem like magic...

- and not necessarily benign magic, at that!
- it's unclear what's happening inside
- leads to anthropomorphizing
- but they're the opposite of us...

they're 100% understandable

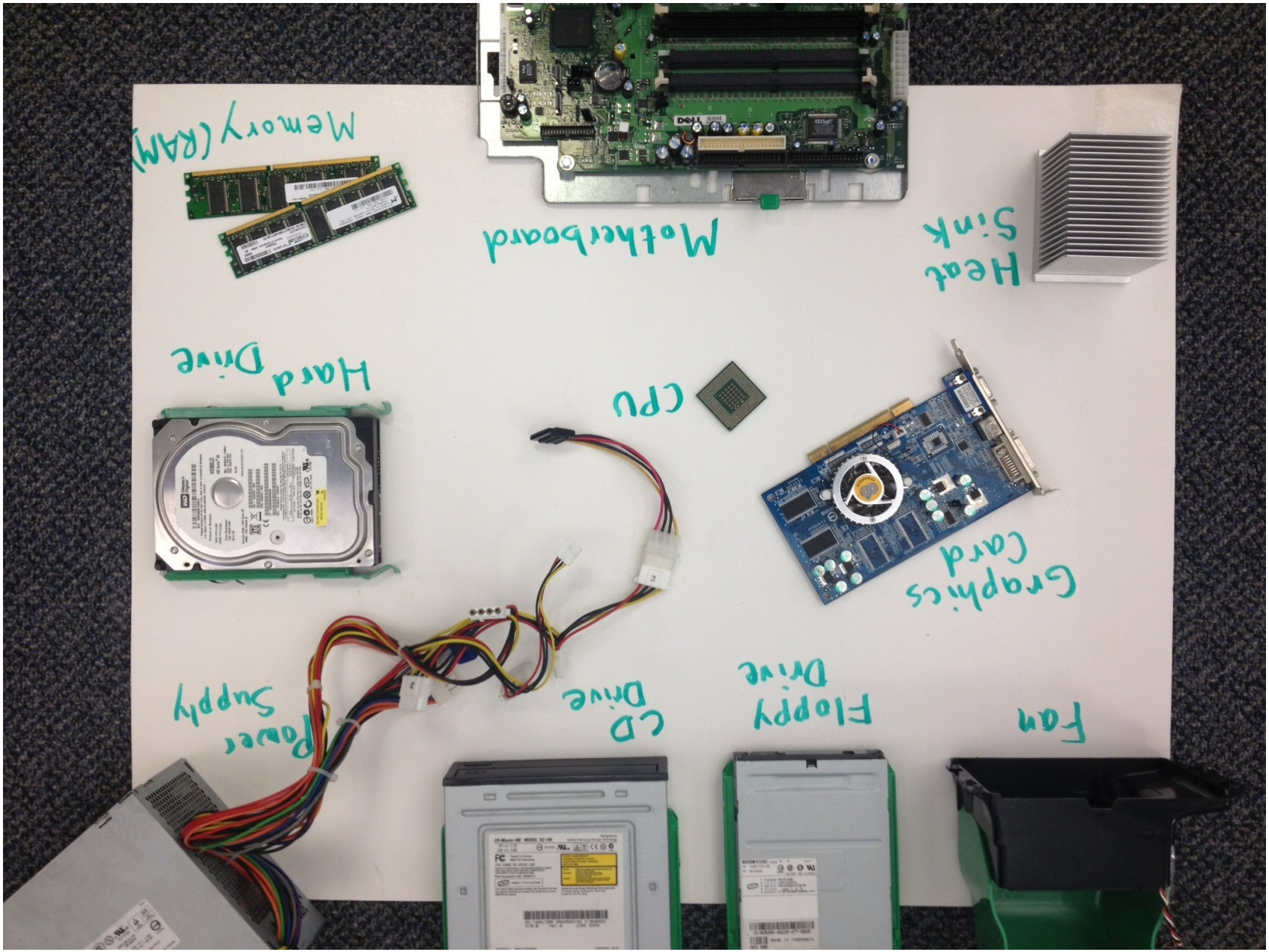
How do we chip away at our defenses?

Computer Disassembly

Use a search engine to look up the parts of a computer - while taking it apart.

Create a poster with the parts and (briefly) what each part does.

Here's an example...



...you should get something like these computer entrails...

1) Find these components in your computer
2) Remove and/or identify each
3) Label each with <u>what it does</u> on your poster

- Memory (RAM)
- Hard Drive
- Transistor
- Motherboard
- Power Supply
- Capacitor
-

- Processor (CPU)
- CD or DVD Drive
- Graphics Card
- LED
- Heat Sink
- Crystal Oscillator
-

4) add two more part names that you find (or know)
5) Finally, take a picture and upload it to our Google doc!

Other parts of computers that you won't find *IN* the computer

- Speakers
- Monitor
- Modem
- Keyboard
- Operating System
- Mouse

Which one of these is not like the others?

How do these fit together? (an abstract picture)

If we make an internet search, in what order would these parts be involved in that process?

___ Keyboard
___ Monitor: show results
___ Graphics Card
___ Memory
___ Processor
___ Network cable or wifi
___ Hard Drive

One answer...

If we make an internet search, in what order would these parts be involved in that process?

on the whiteboard...

True or False?

__ If I get a computer with a better processor, I'll be able to download faster.

__ The larger my processor's clock speed, the faster my computer will be.

__ The more space my computer has, the faster it'll run a lot of programs at once.

__ If I add another processor to this computer, it will run Minecraft with less lag.

__ If I'm running out of space on my computer, I need to get more memory.

Scratch: Movement

Day 1, Session 4

Move!

The move block makes the sprite move **forward**. It has 1 variable:

1. You can change how many **steps** the sprite takes.

The stage is **480** steps wide (-240 to +240) and **360** steps tall (-180 to +180).

The sprite will move in the direction it is **facing**.

Turn!

The turn blocks make the sprite turn **clockwise** or **counterclockwise**. They each have 1 variable:

1. You can change how many **degrees** the sprite turns.

Movement Practice

Can you make your sprite move in a triangle?

In a square?

Pen down!

Pen blocks can be found in the pen **tab**.

The pen down block makes the sprite **draw a line** as it moves.

The sprite will continue to draw a line until you have it lift the pen back **up**.

Pen up!

The pen up block makes the sprite **stop** drawing lines as it moves.

This does NOT make the lines you've already drawn **disappear**.

Clear!

The clear block **erases** all of the pen lines that have already been drawn.

Set pen color!

The set pen color block changes the color of **the line** the sprite draws when the pen is down.

Clicking on the color box will let you choose the color you want from a **palette**.

Set pen size!

The set pen size block changes the **thickness** of the line the sprite draws as it moves. It has 1 variable:

1. You can change how thick the line is. A **bigger** number makes a wider line.

Show and Hide!

If you do not want to see the sprite on the stage, use the hide block to make it **invisible**. The sprite will still move and draw lines while it is invisible.

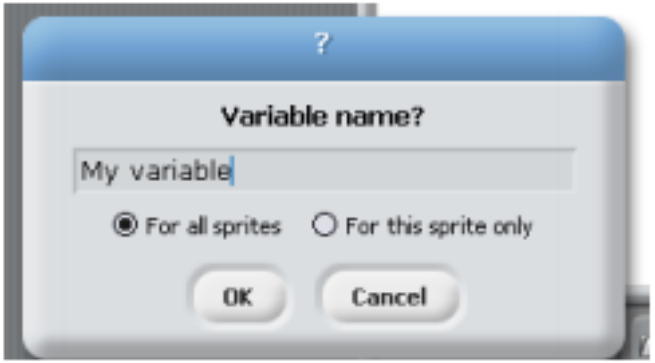
If you want to make the sprite **visible** again, use the show block.

Both the show and hide blocks can be found in the "**Looks**" tab.

Making a New Variable!

To make a new variable:

1. Click on the "**Variables**" tab
2. Click "**Make a variable**"
3. Give your variable a **name**



Making a New Variable, continued!

You now have **new blocks** to use with your variable!

You can add your variable to **any block** that has a variable.

Using Your Variable!

You can change your variable using the **set variable** and **change variable** blocks, just like you did with tempo and volume.

Remember to use the **checkbox** to keep track of what your variable is at any time.

Choose Your Challenges

Pick a challenge from the worksheets and implement it.

Bugs

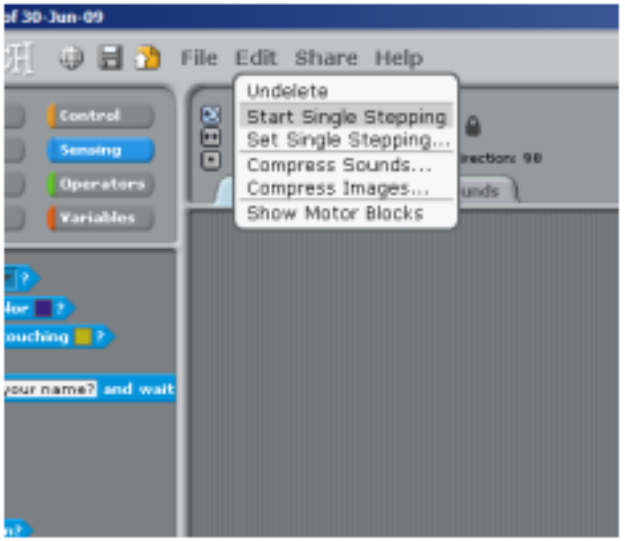
A **bug** is a **mistake** in a computer program that makes it not work the way it is **supposed to**.

If a computer program (or **script**) doesn't work the way it is supposed to, that is almost always because of **human error**.

Debugging

Debugging is the process of **fixing** the bugs in a computer program so that it works the way it is supposed to.

Using Single Stepping



If your script, or program, isn't working the way you expect it to, a good debugging **strategy** can be to run the script in "**single stepping**" mode. This will cause Scratch to run the script more slowly and **highlight** each block as it is run. This can help you see which part of the script is causing the problem.

To start single stepping, go to the **Edit** menu and select "Start Single Stepping".

Debugging Practice

Use the worksheet to practice debugging.

Scratch: *thoughts?*

Could Scratch be a compelling introduction to programming for middle-school students?

Scratch: *thoughts?*

Could Scratch be a compelling introduction to programming for middle-school students?

DIGITAL DOMAIN

Computer Science for the Rest of Us

By RANDALL STROSS
Published: March 31, 2012

READING, writing and — refactoring code?

Marie desJardins, a computer science professor at the University of Maryland, Baltimore County, says her department uses Scratch in its "Introduction to Computers and Programming" course, in which students can try a few basic concepts. About 25 percent of the semester is spent on programming.

Explaining why Scratch is used at the college level, she says that all students arrive on campus having taken high school classes in English, math, biology and so on, but that many have not taken a computer science class.

Michael Littman, who leads the computer science department at Rutgers University, agrees. "Computational thinking should have been covered in middle school, and it isn't," he says. "So we in the C.S. department must offer the equivalent of a remedial course."

It's increasingly popular in colleges' CS1 courses… !

Thanks & see you tomorrow!

4-7 days spiraling

MyCS

Middle-years Computer Science

Tomorrow's plan

Day 1	CS: what and why?
	Scratch ~ sound
	What is a computer?
	Scratch ~ movement
Day 2	CS as problem-solving
	Encoding data
	Scratch ~ coordinates
Day 3	*Algorithms* ?!
	Scratch ~ if/else
	Limits of computers
	Scratch ~ projects
Day 4	Using the web ***well***
	Web: HTML/CSS
	Web safety/security
	Web: Javascript
Day 5	Games and extras