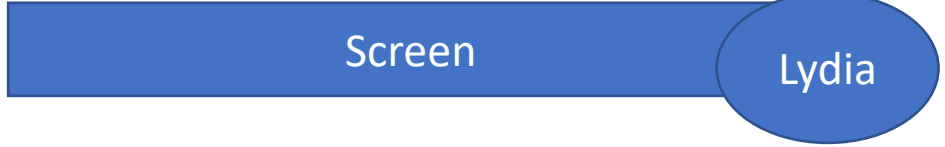
Design Project Kickoff

Prof. Lydia Chilton
COMS 4170
21 March 2022



p1

Empty row

p2

Empty row

p5

Empty row

p6

Empty row

p7

Empty row

p9

Empty row

p10

Empty row

p11

Empty row

p13

Empty row

p15

Empty row

p16

Empty row

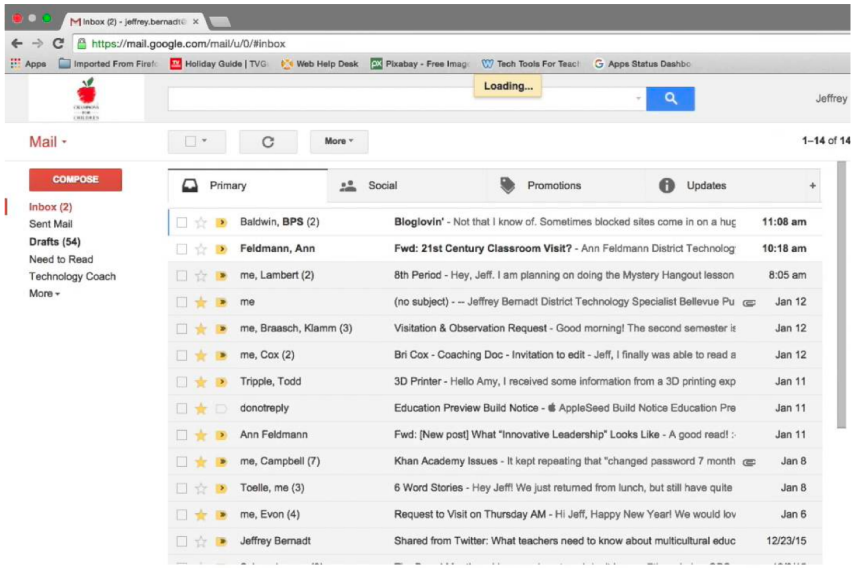
p17

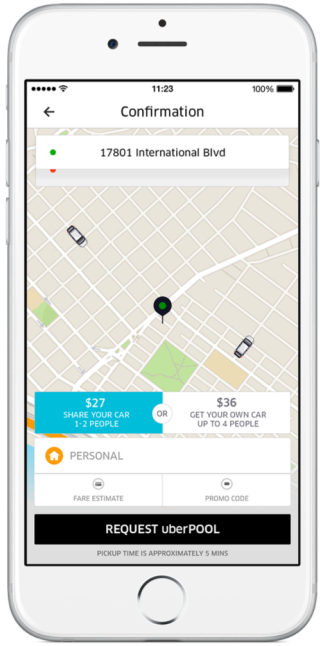
NOTE: p18 and p19 should sit anywhere for now (empty rows are ok).

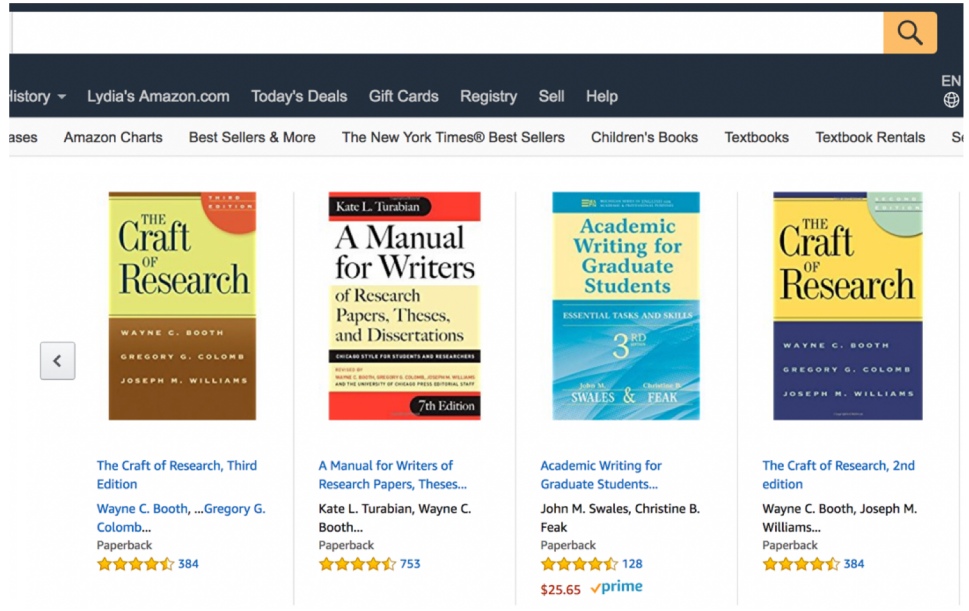
Homework 7 is a big achievement!

You are now a user interface programmer!

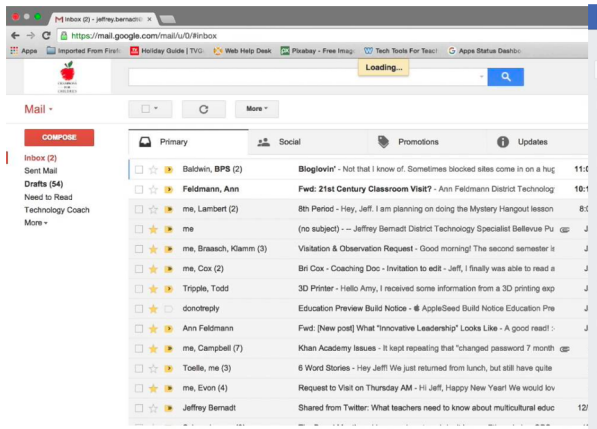
The main goal of many websites is to **interact with data.**

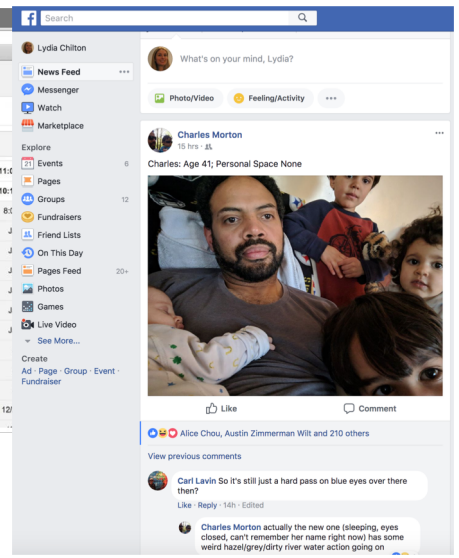


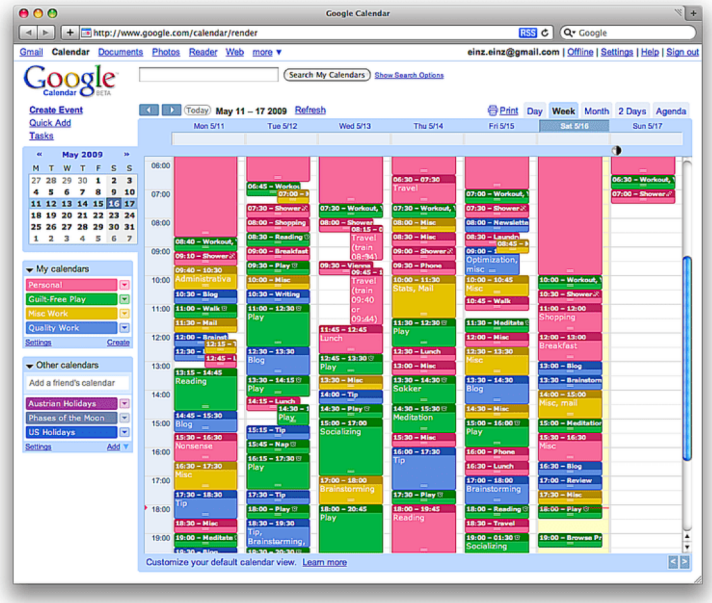


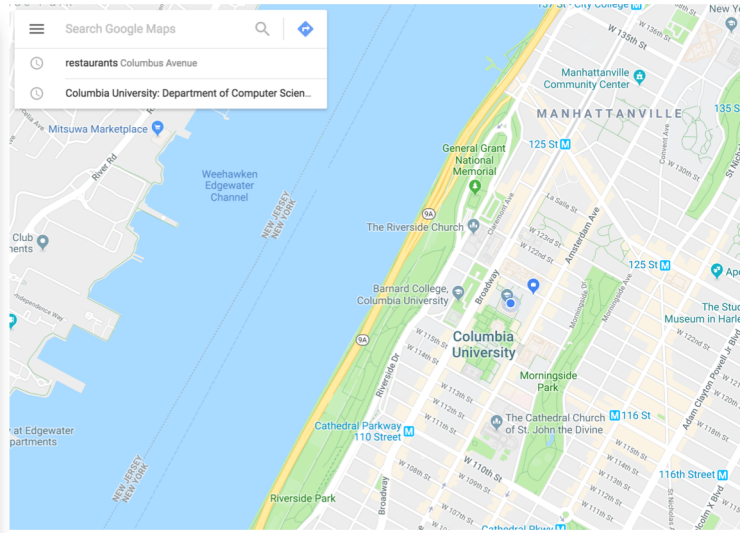


You can now make a working prototype of a most websites







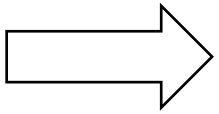


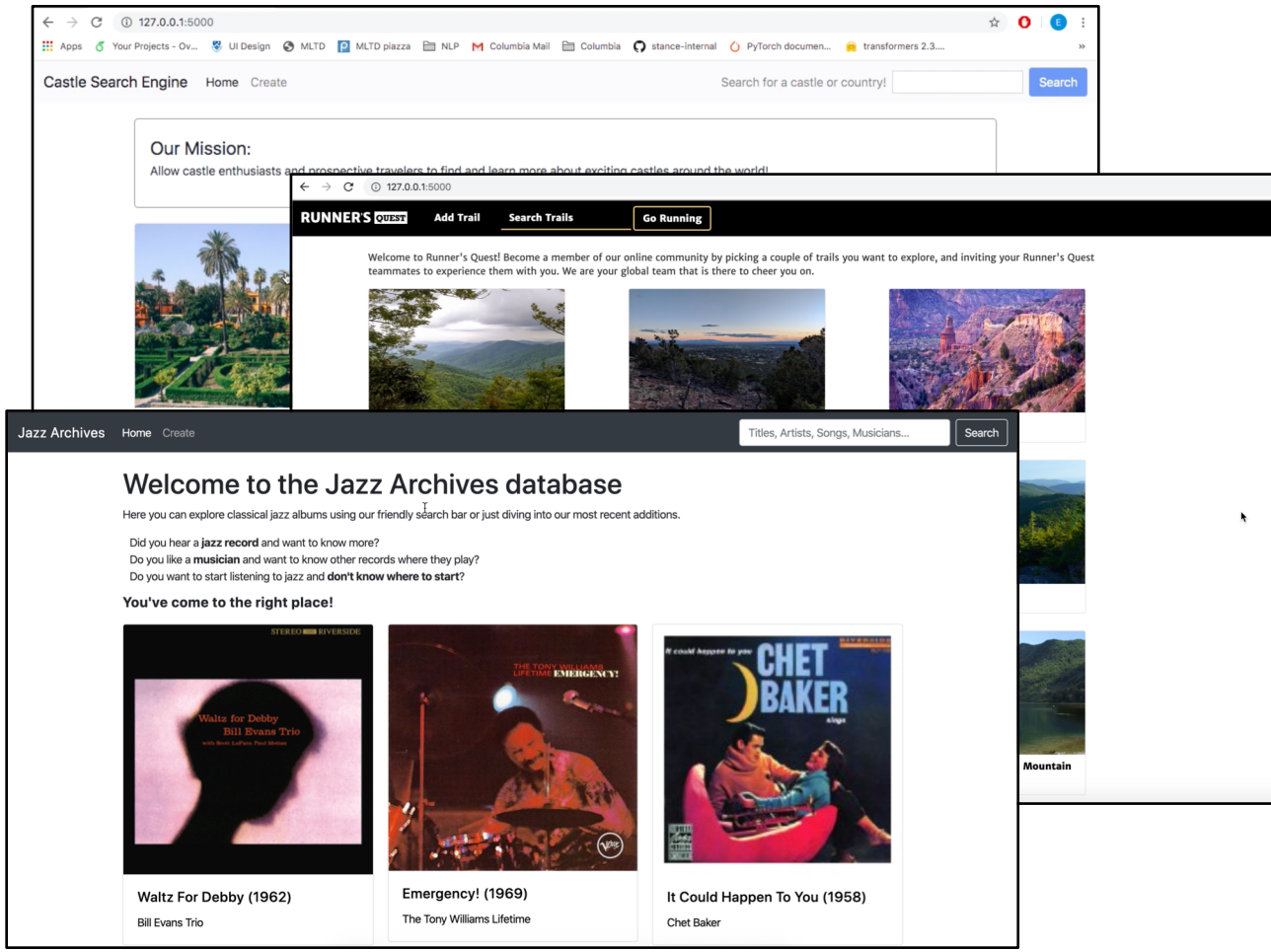
Given specifications, you can create interactions

Part 1 – Usable Functionality:

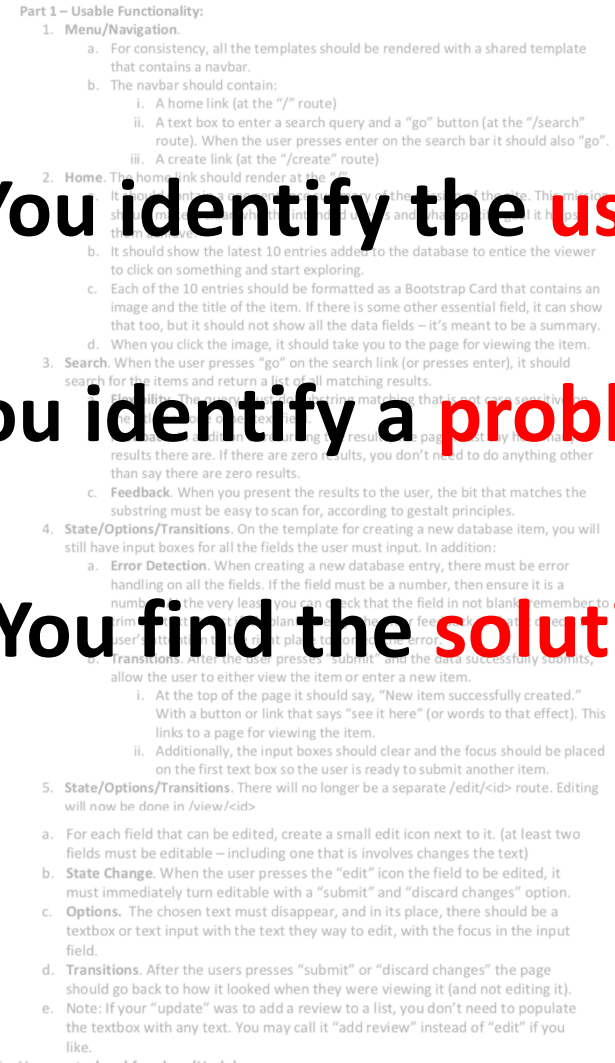
1. **Menu/Navigation**.
   a. For consistency, all the templates should be rendered with a shared template that contains a navbar.
   b. The navbar should contain:
      i. A home link (at the "/" route)
      ii. A text box to enter a search query and a "go" button (at the "/search" route). When the user presses enter on the search bar it should also "go".
      iii. A create link (at the "/create" route)
2. **Home**. The home link should render at the "/".
   a. It should contain a one sentence summary of the mission of the site. This mission should make it clear who the intended user is and what specific goal it helps them achieve.
   b. It should show the latest 10 entries added to the database to entice the viewer to click on something and start exploring.
   c. Each of the 10 entries should be formatted as a Bootstrap Card that contains an image and the title of the item. If there is some other essential field, it can show that too, but it should not show all the data fields – it's meant to be a summary.
   d. When you click the image, it should take you to the page for viewing the item.
3. **Search**. When the user presses "go" on the search link (or presses enter), it should search for the items and return a list of all matching results.
   a. **Flexibility**. The query must do substring matching that is not case sensitive on the title and one other text field.
   b. **Feedback**. In addition to returning the results, the page must say how many results there are. If there are zero results, you don't need to do anything other than say there are zero results.
   c. **Feedback**. When you present the results to the user, the bit that matches the substring must be easy to scan for, according to gestalt principles.
4. **State/Options/Transitions**. On the template for creating a new database item, you will still have input boxes for all the fields the user must input. In addition:
   a. **Error Detection**. When creating a new database entry, there must be error handling on all the fields. If the field must be a number, then ensure it is a number. At the very least, you can check that the field in not blank (remember to trim the text to test if it's blank). Design the error feedback so that it directs the user's attention to the right place to correct the error.
   b. **Transitions**. After the user presses "submit" and the data successfully submits, allow the user to either view the item or enter a new item.
      i. At the top of the page it should say, "New item successfully created." With a button or link that says "see it here" (or words to that effect). This links to a page for viewing the item.
      ii. Additionally, the input boxes should clear and the focus should be placed on the first text box so the user is ready to submit another item.
5. **State/Options/Transitions**. There will no longer be a separate /edit/<id> route. Editing will now be done in /view/<id>

   a. For each field that can be edited, create a small edit icon next to it. (at least two fields must be editable – including one that is involves changes the text)
   b. **State Change**. When the user presses the "edit" icon the field to be edited, it must immediately turn editable with a "submit" and "discard changes" option.
   c. **Options.** The chosen text must disappear, and in its place, there should be a textbox or text input with the text they way to edit, with the focus in the input field.
   d. **Transitions**. After the users presses "submit" or "discard changes" the page should go back to how it looked when they were viewing it (and not editing it).
   e. Note: If your "update" was to add a review to a list, you don't need to populate the textbox with any text. You may call it "add review" instead of "edit" if you like.
6. **User control and freedom (Undo)**.
   a. The user should no longer be able to delete entire database items from the search page.





The next step is to become a user interface **designer**.



Part 1 – Usable Functionality:

1. **Menu/Navigation**.
   a. For consistency, all the templates should be rendered with a shared template that contains a navbar.
   b. The navbar should contain:
      i. A home link (at the "/" route)
      ii. A text box to enter a search query and a "go" button (at the "/search" route). When the user presses enter on the search bar it should also "go".
      iii. A create link (at the "/create" route)
2. **Home**. The home link should render at the
   b. It should show the latest 10 entries added to the database to entice the viewer to click on something and start exploring.
   c. Each of the 10 entries should be formatted as a Bootstrap Card that contains an image and the title of the item. If there is some other essential field, it can show that too, but it should not show all the data fields – it's meant to be a summary.
   d. When you click the image, it should take you to the page for viewing the item.
3. **Search**. When the user presses "go" on the search link (or presses enter), it should search for the items and return a list of all matching results.
   results there are. If there are zero results, you don't need to do anything other than say there are zero results.
   c. **Feedback**. When you present the results to the user, the bit that matches the substring must be easy to scan for, according to gestalt principles.
4. **State/Options/Transitions**. On the template for creating a new database item, you will still have input boxes for all the fields the user must input. In addition:
   a. **Error Detection**. When creating a new database entry, there must be error handling on all the fields. If the field must be a number, then ensure it is a
   b. **Transitions**. After the user presses "submit" and the data successfully submits, allow the user to either view the item or enter a new item.
      i. At the top of the page it should say, "New item successfully created." With a button or link that says "see it here" (or words to that effect). This links to a page for viewing the item.
      ii. Additionally, the input boxes should clear and the focus should be placed on the first text box so the user is ready to submit another item.
5. **State/Options/Transitions**. There will no longer be a separate /edit/<id> route. Editing will now be done in /view/<id>

   a. For each field that can be edited, create a small edit icon next to it. (at least two fields must be editable – including one that is involves changes the text)
   b. **State Change**. When the user presses the "edit" icon the field to be edited, it must immediately turn editable with a "submit" and "discard changes" option.
   c. **Options.** The chosen text must disappear, and in its place, there should be a textbox or text input with the text they way to edit, with the focus in the input field.
   d. **Transitions**. After the users presses "submit" or "discard changes" the page should go back to how it looked when they were viewing it (and not editing it).
   e. Note: If your "update" was to add a review to a list, you don't need to populate the textbox with any text. You may call it "add review" instead of "edit" if you like.

You identify the user

You identify a problem

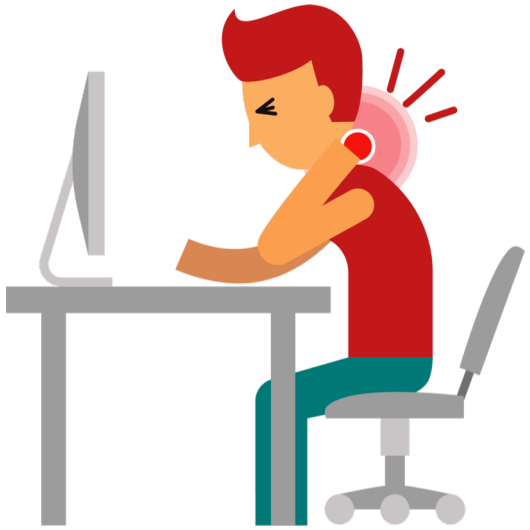
You find the solution

What is design?

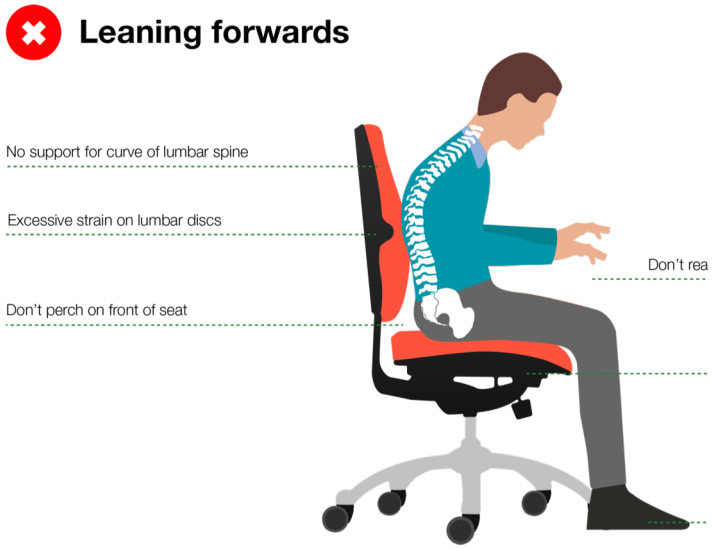
“Design is a plan for arranging elements to accomplish a particular purpose.”

– Charles Eames

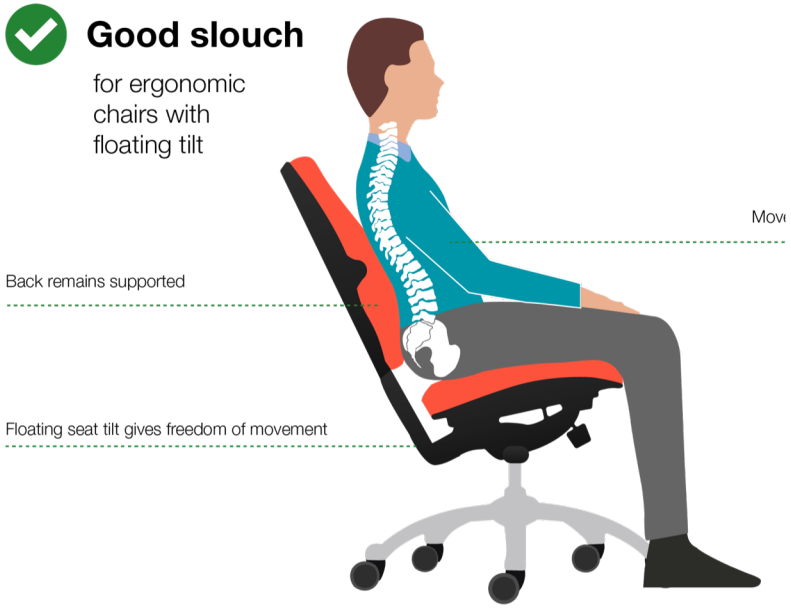
Design is a iterative progress where you work with users to identify and solve their problems.



Sitting all day hurts!



Why does it hurt?
How do people sit?



What does good sitting look like?

Identify Users needs:

For people who sit all day in an office, alleviate back pain by designing a chair that supports the lower back.

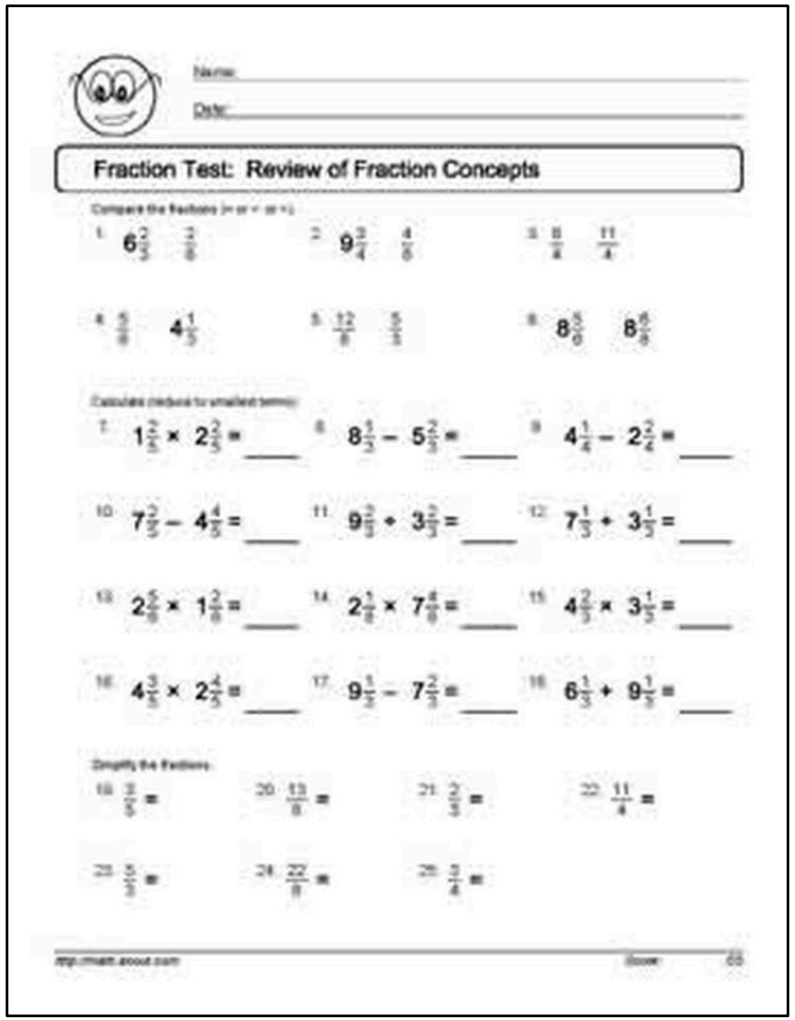
Test solutions on users:

The key to design is to iterate based on user feedback.

Let me tell you a story

This is Nadia. She’s 11. She lives in Houston.

Nadia is struggling with fractions at school



Name:

Date:

Fraction Test: Review of Fraction Concepts

Compare the fractions (> or < or =).

1. $6\frac{2}{3}$ $\frac{2}{6}$
2. $9\frac{3}{4}$ $\frac{4}{8}$
3. $\frac{8}{4}$ $\frac{11}{4}$
4. $\frac{5}{8}$ $4\frac{1}{5}$
5. $\frac{12}{8}$ $\frac{5}{3}$
6. $8\frac{5}{6}$ $8\frac{6}{9}$

Calculate (reduce to smallest terms):

7. $1\frac{2}{5} \times 2\frac{2}{3} =$ ____
8. $8\frac{1}{3} - 5\frac{2}{3} =$ ____
9. $4\frac{1}{4} - 2\frac{2}{4} =$ ____
10. $7\frac{2}{5} - 4\frac{4}{5} =$ ____
11. $9\frac{3}{8} + 3\frac{2}{8} =$ ____
12. $7\frac{1}{3} + 3\frac{1}{3} =$ ____
13. $2\frac{5}{6} \times 1\frac{2}{6} =$ ____
14. $2\frac{1}{8} \times 7\frac{4}{6} =$ ____
15. $4\frac{2}{3} \times 3\frac{1}{3} =$ ____
16. $4\frac{3}{5} \times 2\frac{4}{5} =$ ____
17. $9\frac{1}{3} - 7\frac{2}{3} =$ ____
18. $6\frac{1}{3} + 9\frac{1}{3} =$ ____

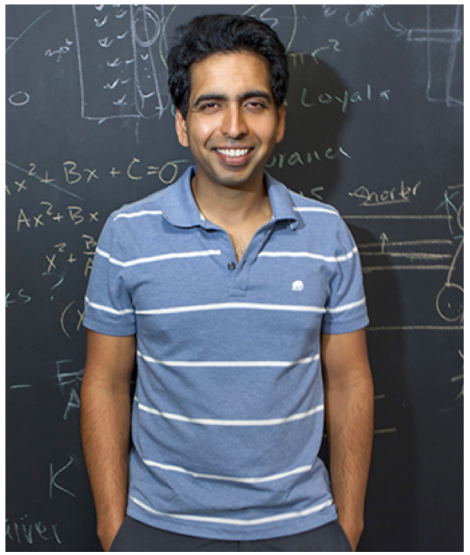
Simplify the fractions:

19. $\frac{3}{5} =$
20. $\frac{13}{8} =$
21. $\frac{2}{3} =$
22. $\frac{11}{4} =$
23. $\frac{5}{3} =$
24. $\frac{22}{8} =$
25. $\frac{3}{4} =$

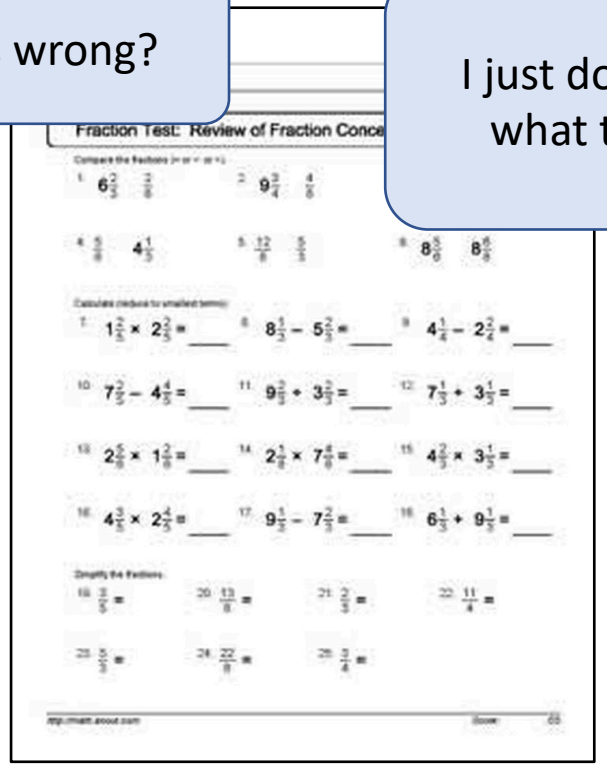
http://math.about.com Score: /25

Her uncle wants to help, but he lives in NYC

He asks about her fractions homework.



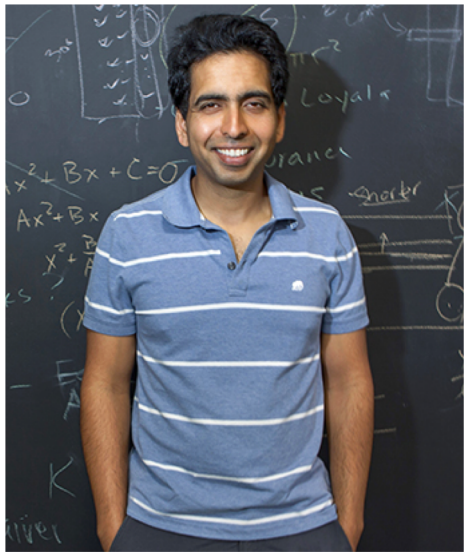
What's wrong?

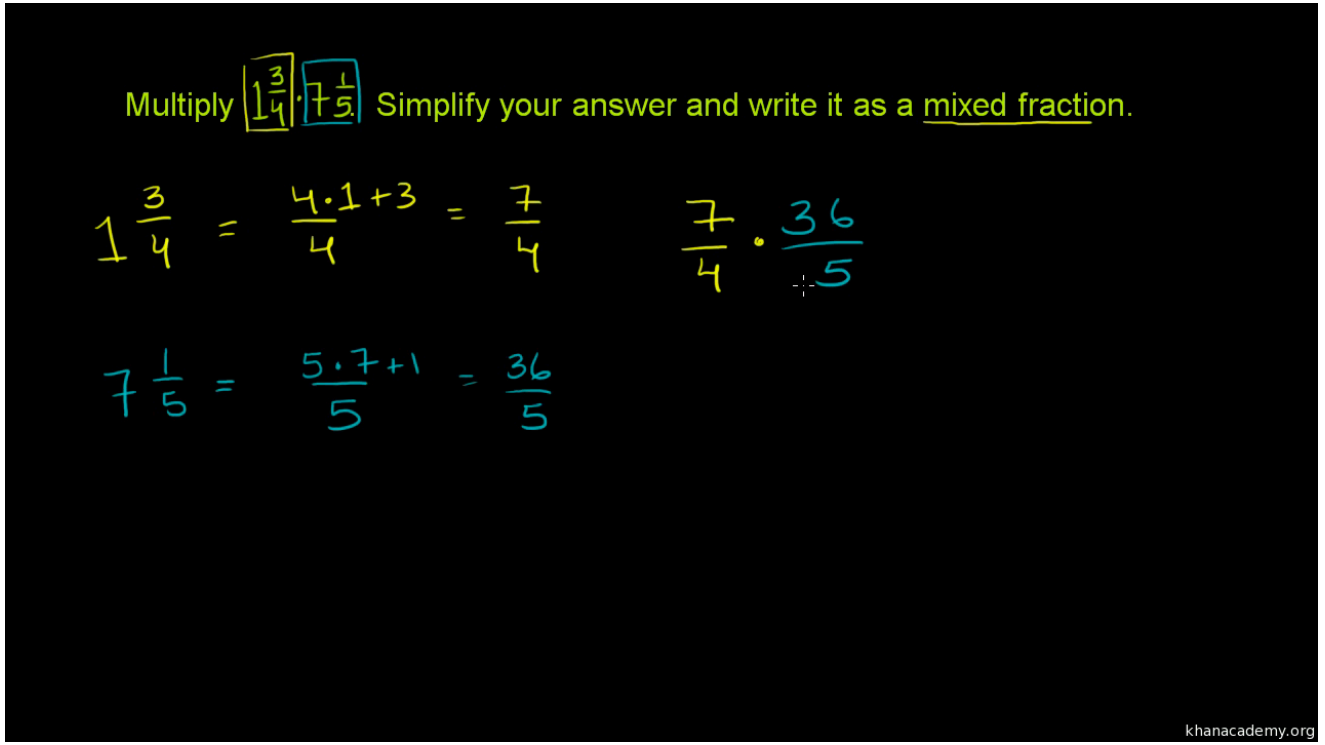


I just don't get what to do.

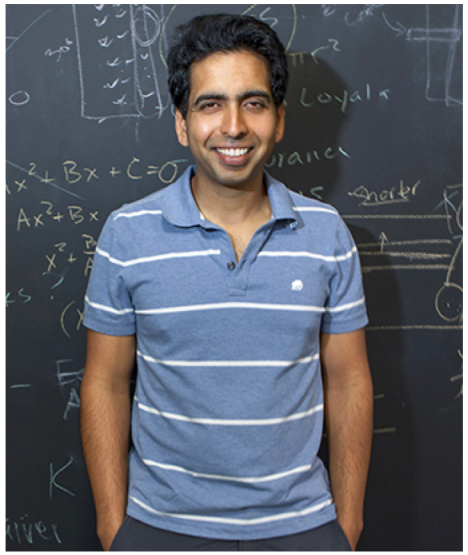
He has the insight that if she **saw more examples**, she could figure out how to solve fractions.

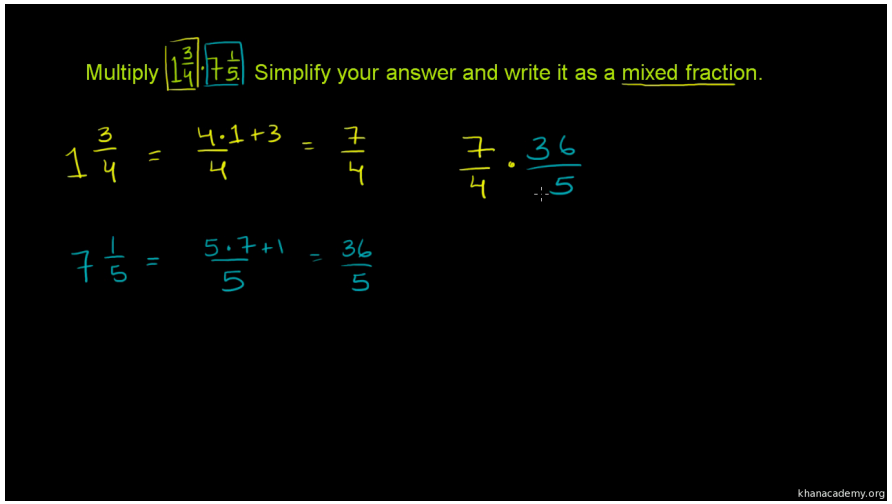
He makes videos and uploads them to YouTube.

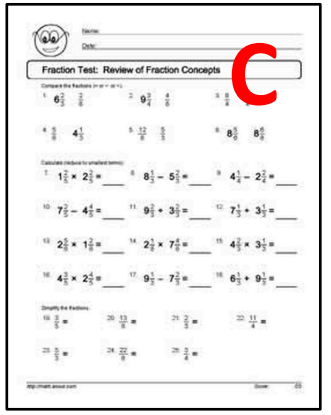




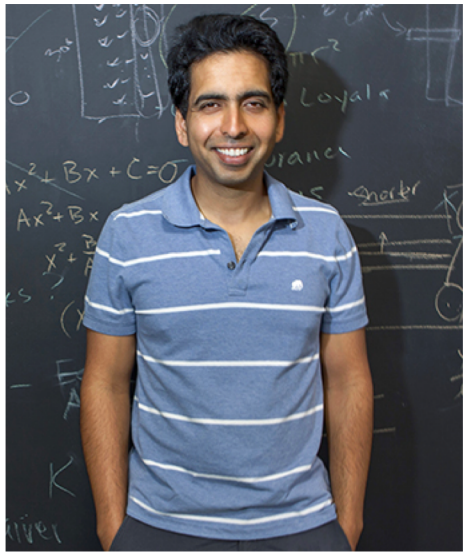
And he shows it to Nadia to see it helps…

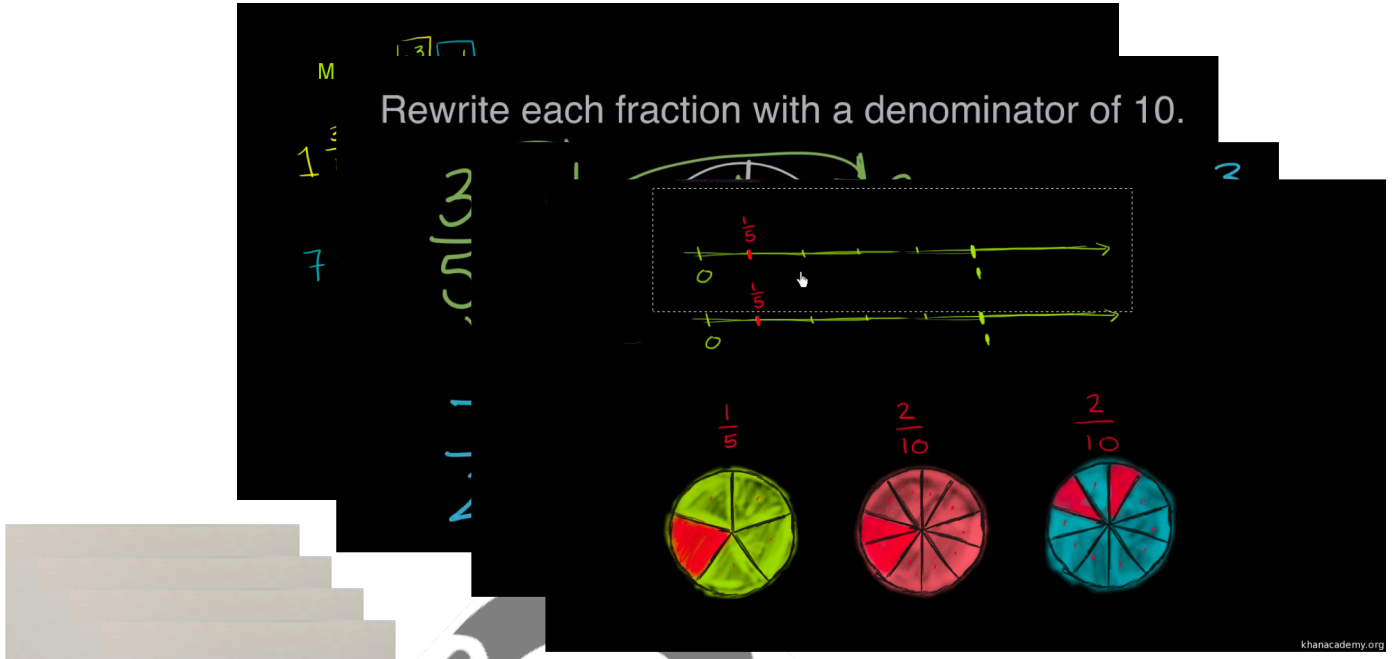


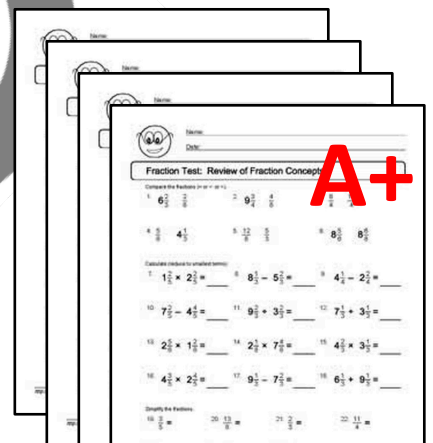




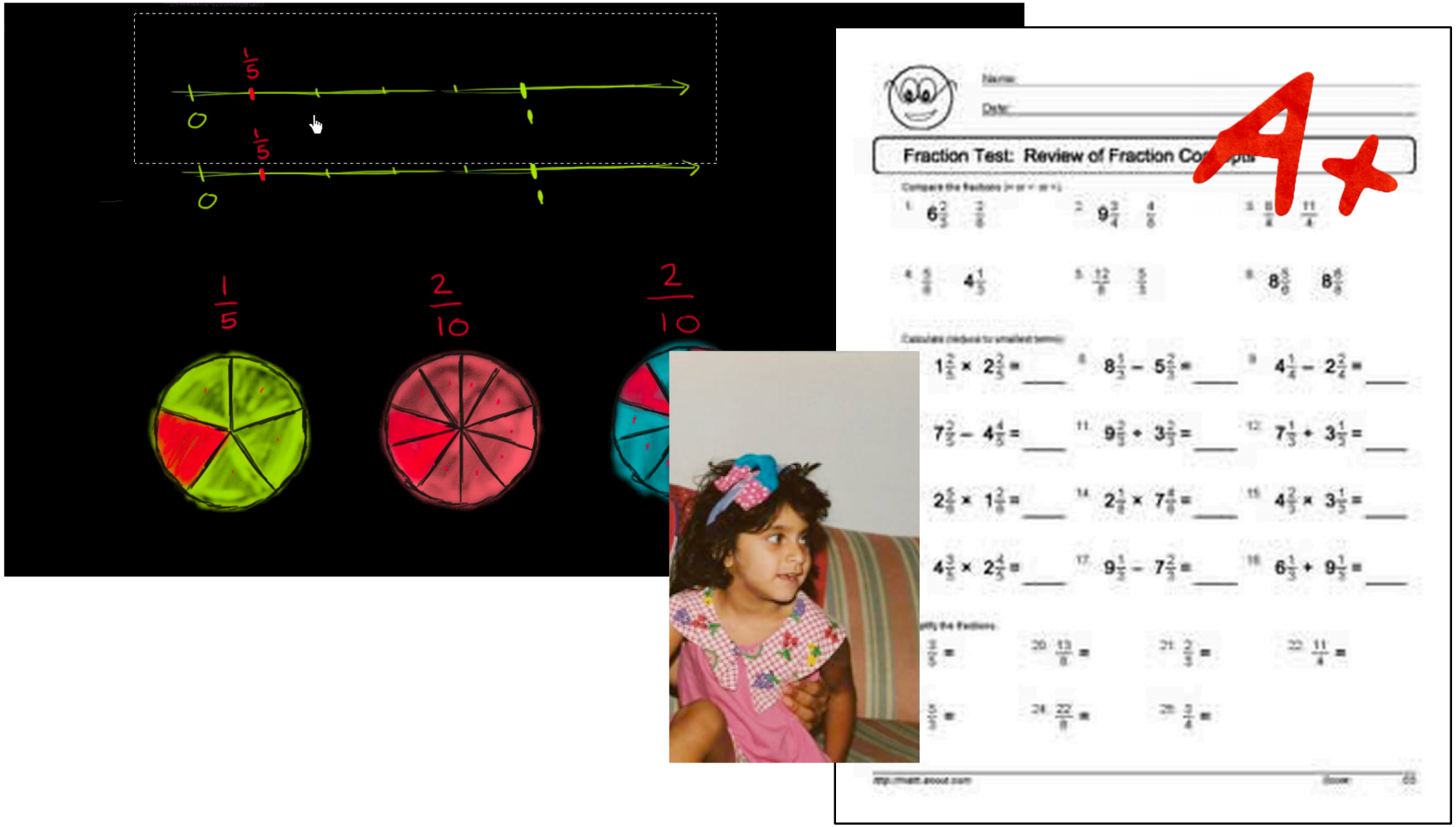
And he improves it again and again



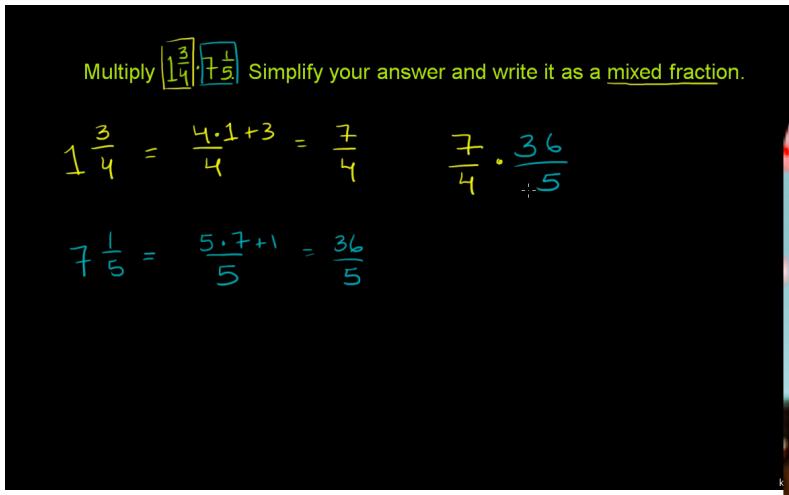




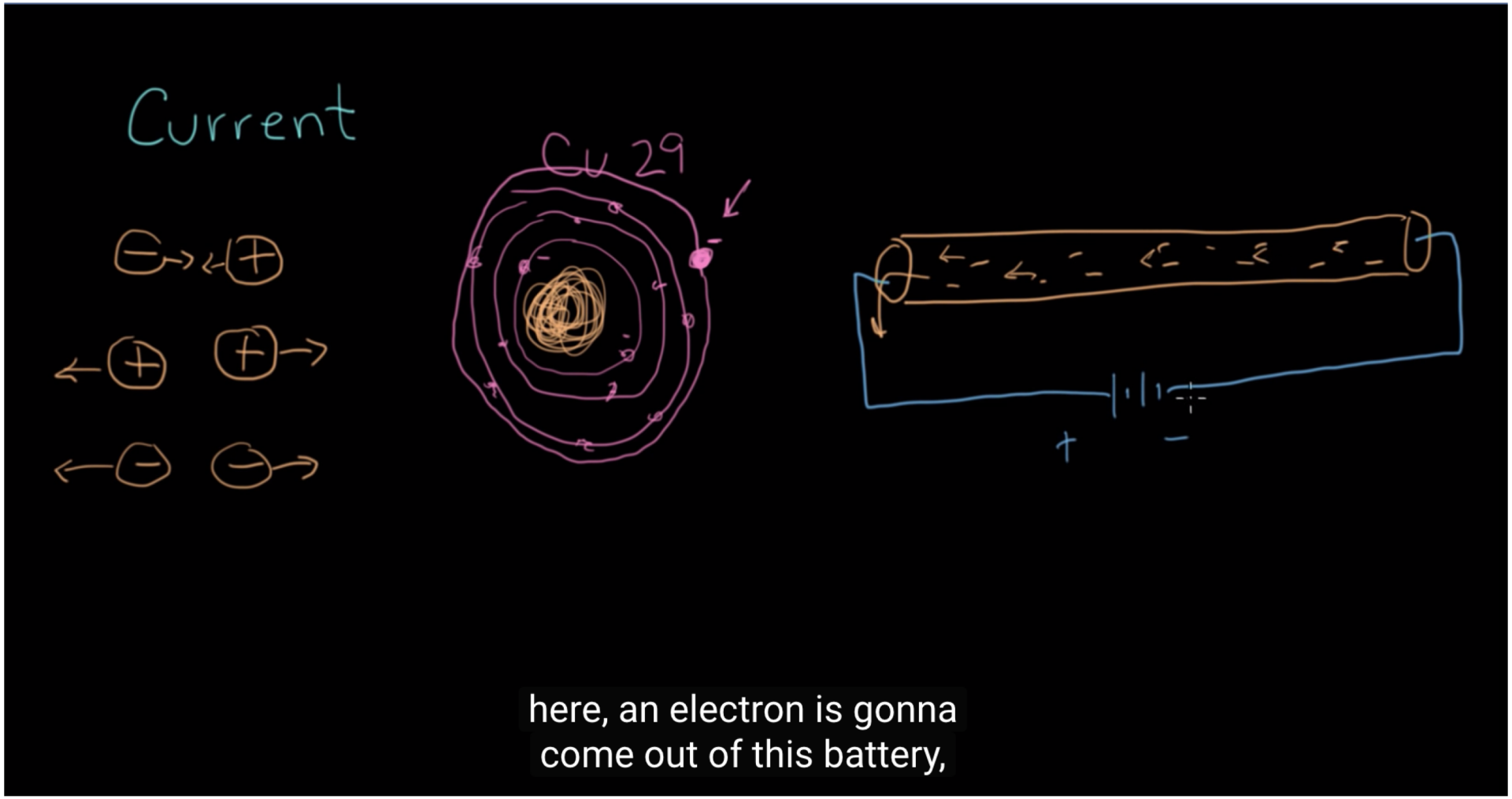
This videos help Nadia. She aces the test!



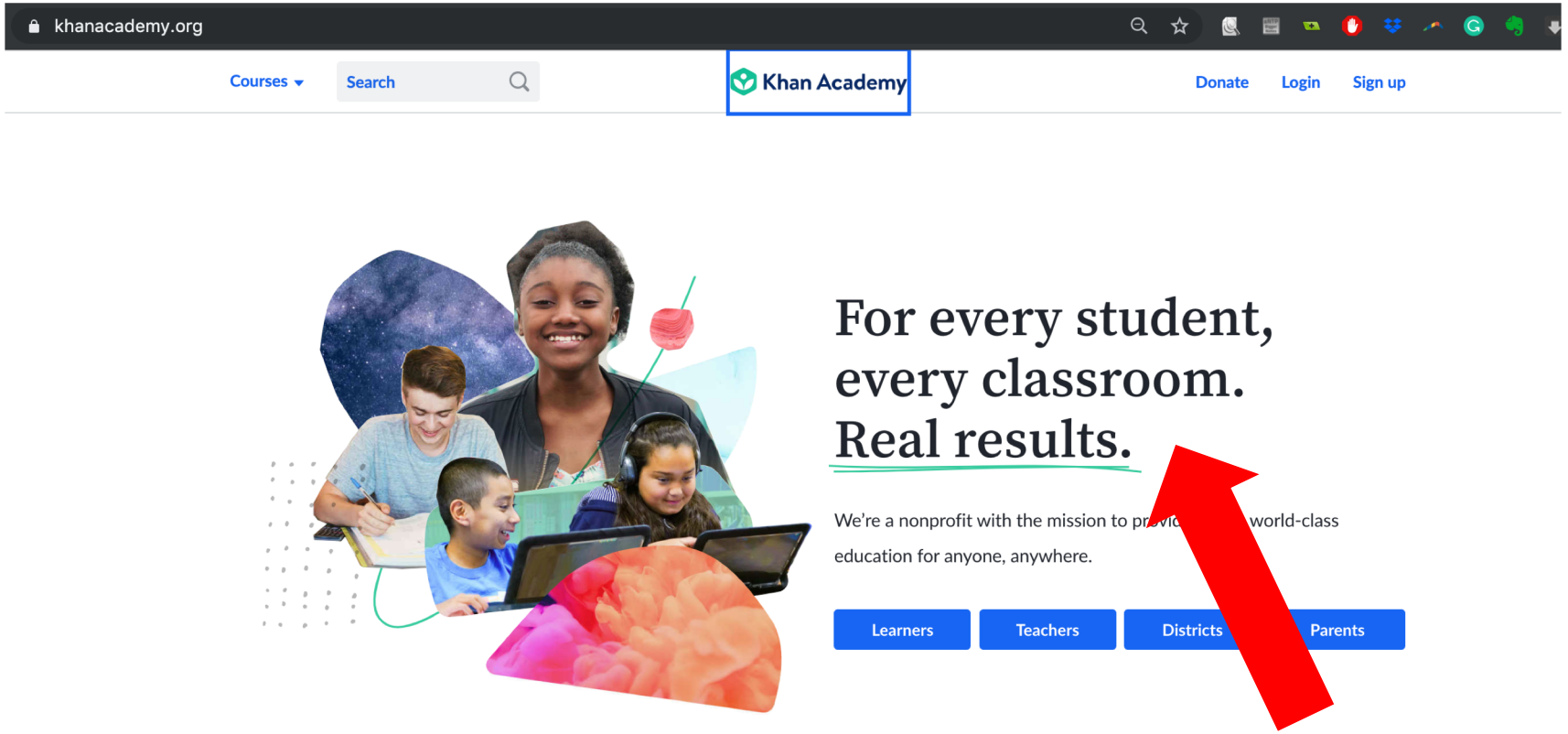
This video helps other students pass their fractions tests, too.



Videos in this format help students learn other topics.

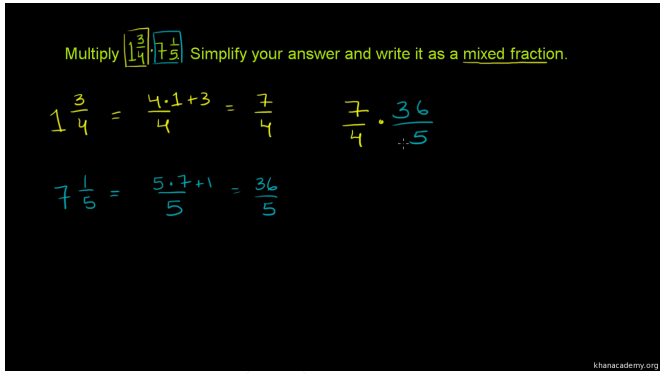


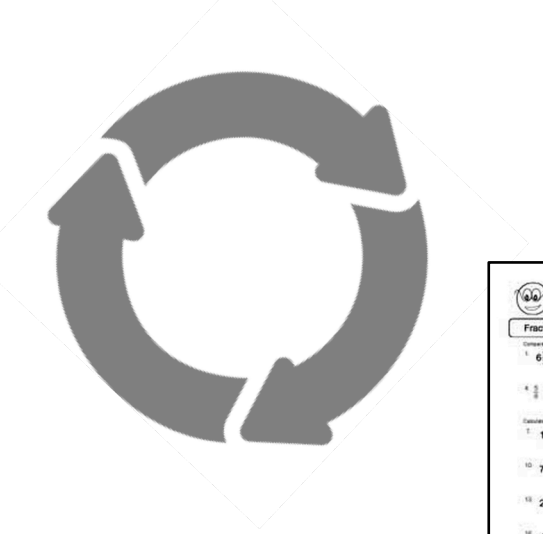
These videos became Khan Academy.

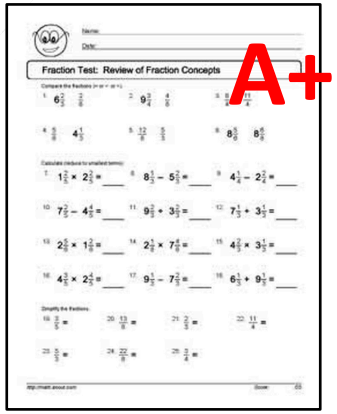


Very general problem

But it started by helping a **person** with a **problem,** and lot of **iteration**







Design Project

Challenge:

- Design and build a **web** application
- That allows a user to **interact** with media
- Within a **domain of your choosing**
- To help a user learn an introductory topic **interactively**
- And help them assess themselves with a **quiz**.
- And keep learning through **feedback** from the quiz.
- In under **10 minutes** total

The **user** is someone in this class

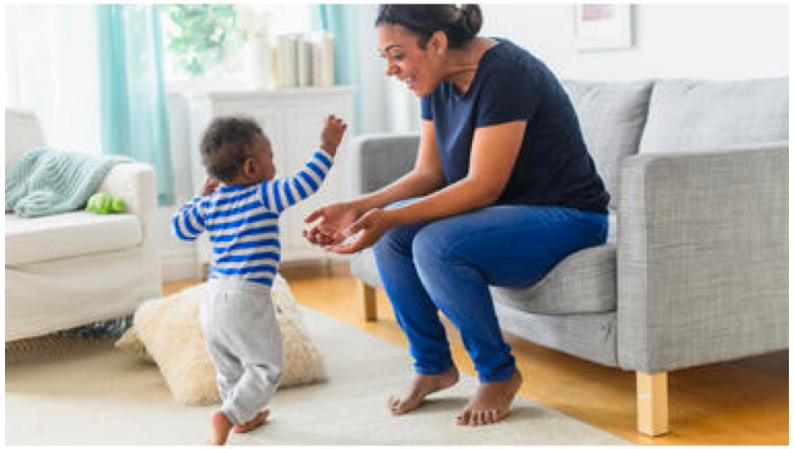
- This way, you can test your designing on people in this class.
- Your TA must also feel like this is something valuable for them to learn.
- Consequently, you **cannot** design for:
  - Kids or teenagers (they aren't in this class)
  - People who only speak Serbian (everyone in this class speaks English)
  - Architecture Majors.

The **topic** may be in a domain of your choice, but it must be focused enough to teach in 10 min.

- Examples of broad domains:
  - chess,
  - basketball,
  - art history,
  - music
- Example focused topics
  - **Chess**: how and when to perform 3 different opening moves in chess for chess beginnings
  - **Basketball**: how to run a pick and roll in basketball for casual NBA fans
  - **Art history**: how to tell impressionist paintings from post-impressionist paintings for ArtHum students
  - **Music**: how to mix a drop swap for aspiring DJs

Design insight for teaching:

- People learn through interaction and feedback, not from reading long dumps of information



Learning to walk

Playing an instrument

```
void loop() {

float humidity = 452;
for( int counter = 0; counter < 1000; ++counter ) {
  digitalWrite(13, HIGH);   // set the LED on
  delay( 200 );             // wait for a second
  digitalWrite(13, LOW);    // set the LED offer
  Serial.pirnt( "Hello " );
  Serial.println( counter );
  delay(200);               // wait for a second
  humidity += 0.5;
}
```

100 %

Error List

2 Errors | 1 Warning | 0 Messages

	Description
1	IntelliSense: class "Serial_" has no member "pirnt"
2	In function 'void loop()'
3	23:10: error: 'class Serial_' has no member named 'pirnt'

Code

Feedback is how we learn

Examples

Lipreading Learn B Learn H Learn L Test

Welcome to **Lipreading**, your site for learning to read lips! Use the navbar above or click the button below to learn the how to lipread different sounds. When you're ready, try a test!

Start With B

How does it fulfill the requirements?

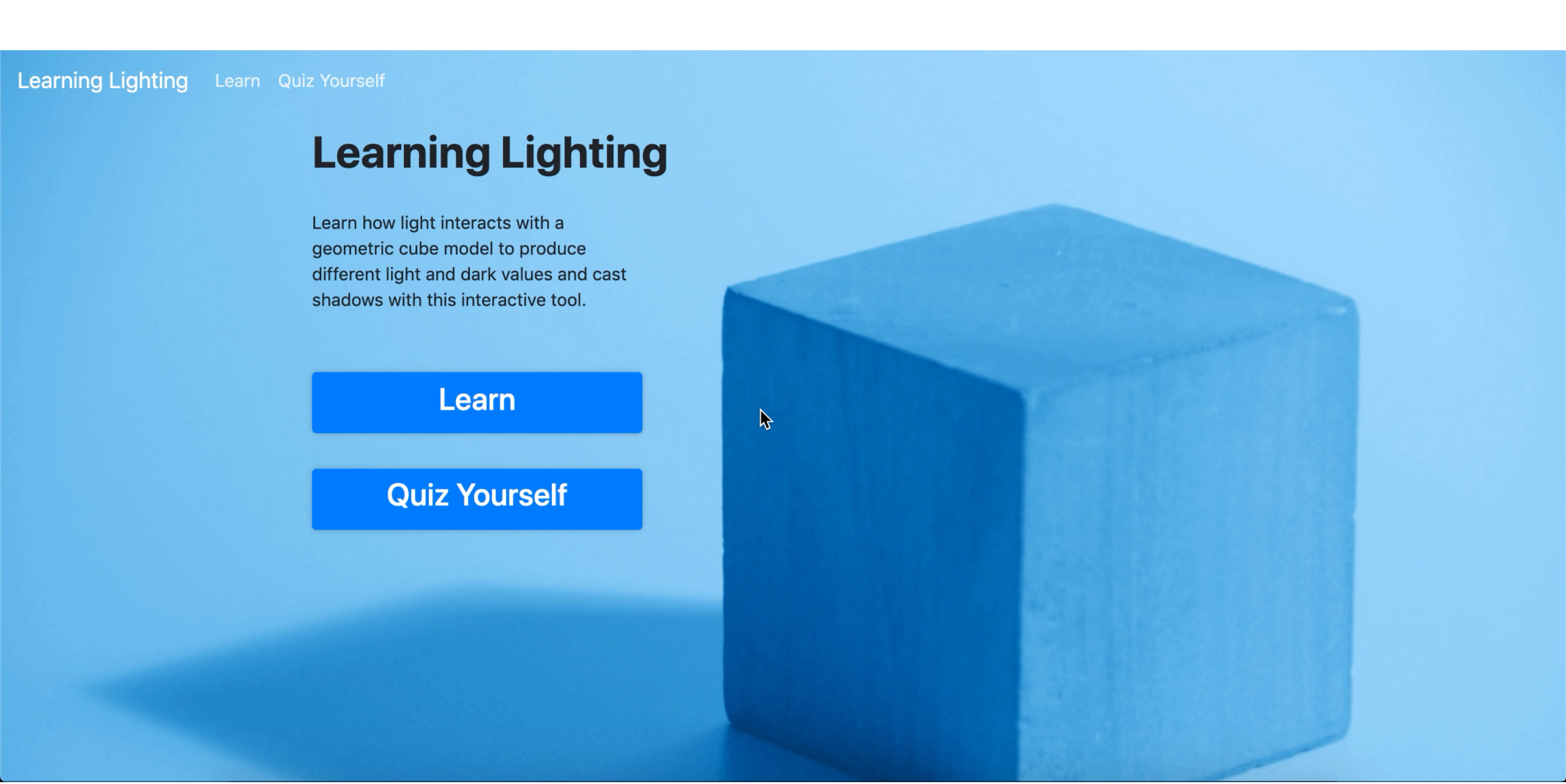
Domain:	Lipreading
Topic:	B, H and L sounds
Media:	Videos
Interaction	Watch videos with and without sound
User:	Would you learning something from this?

Learning Lighting

Learn how light interacts with a geometric cube model to produce different light and dark values and cast shadows with this interactive tool.

Learn

Quiz Yourself

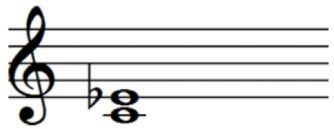


How does it fulfill the requirements?

Domain:	Lighting
Topic:	Lighting from 5 directions
Media:	3d model
Interaction	Click the model
User:	Would you learning something from this?

Minor Third

Step 1: Listen



A Minor Third has 3 half steps between the lower note and the higher note. Try counting the number of lines and spaces between the two notes on the image above.

The "minor" quality indicates that this interval comes from the minor scale of its lower note, and that the interval sounds slightly dissonant or unhappy.

0:00 / 0:06

Step: 1 2 3

How does it fulfill the requirements?

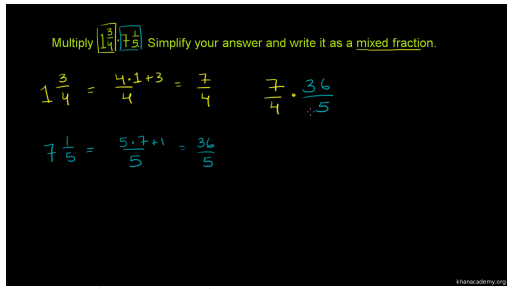
Domain:	Music
Topic:	Identifying intervals
Media:	Piano
Interaction	Play music, press keys on piano
User:	Would you learning something from this?

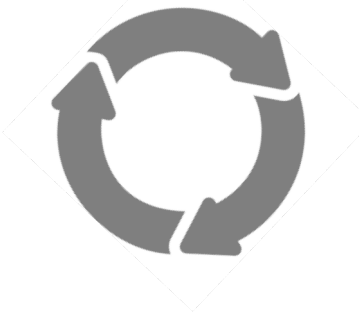
Project Logistics

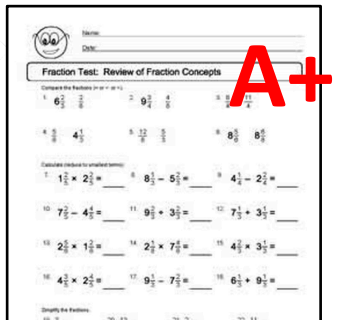
- **Weekly homework** will build up to the final project (5% of grade)
- Final submission is worth **20% of your grade**.
- This project is to be completed **in a group**.
  - You will meet with your TA to receive feedback.

You must iterate based on your TAs feedback.

- There are no right or wrong answers to design problems.
  - But there are better and worse answers.
- A core skill we want you to learn is to iterate based on feedback. Thus, your grade is depending on your making your TA happy.



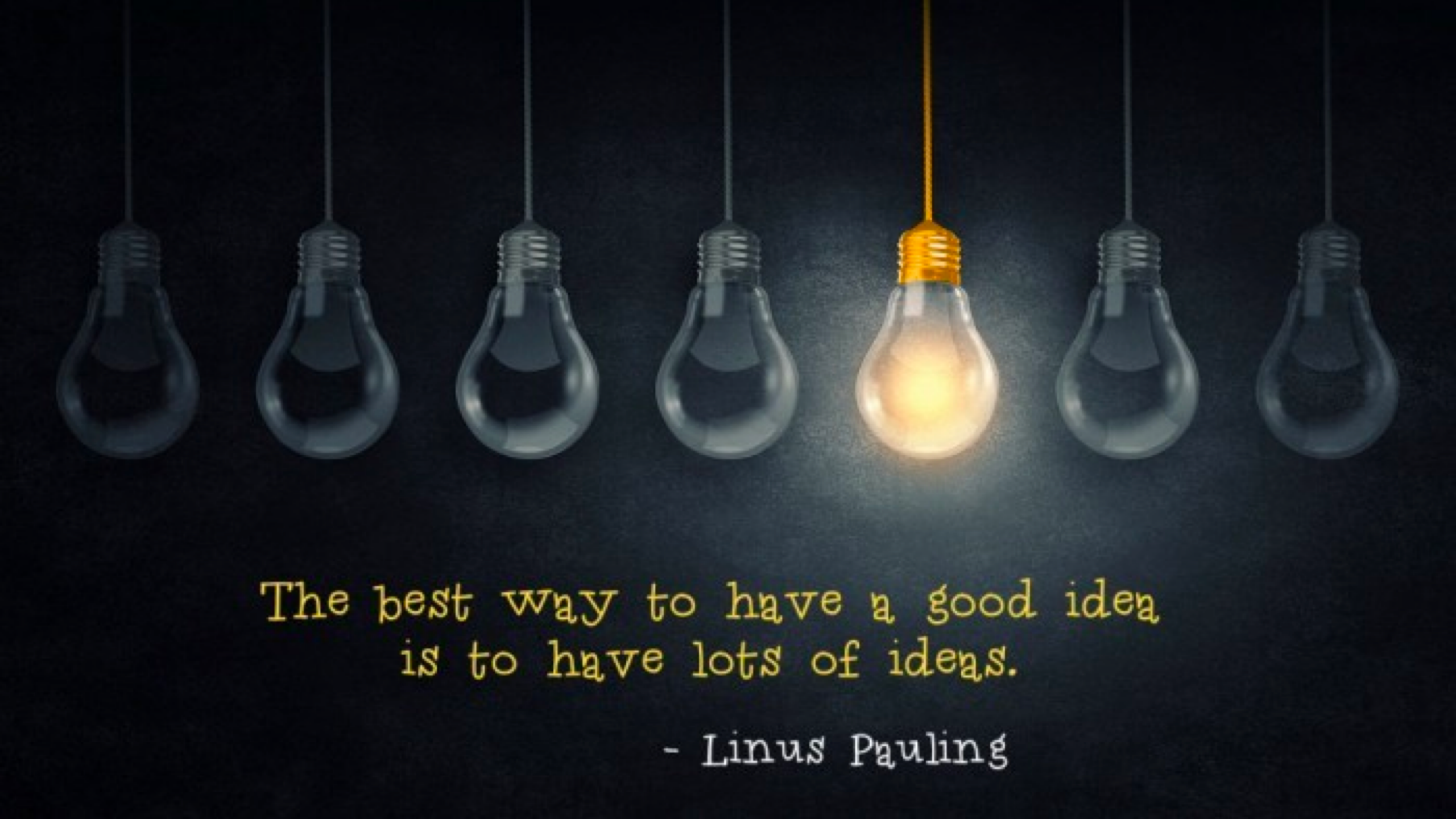




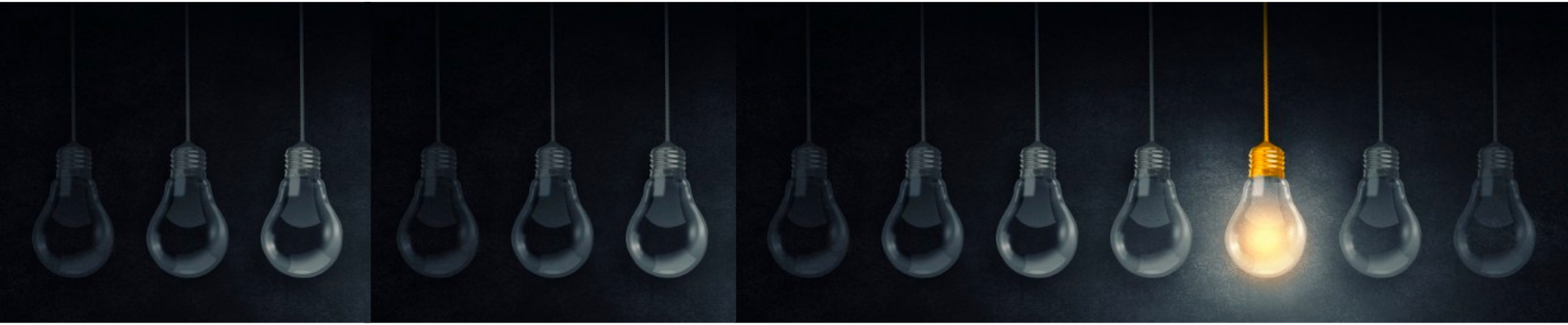
In class now: Brainstorming Domains and Topics

Warm-up (due Wednesday)
Group brainstorming domains and topics

- Meet your group!
- Individual Brainstorm (5 minutes)
  - What are 5 domains that you could teach things to your classmates?
  - For each of those 5 domains, list 5 specific topics that you could teach interactively in under 10 minutes
- Group brainstorm:
  - Compare ideas, iterate!
  - **Users**: Would people in this class learn something from it?
  - **Topic**: is the topic focused enough to teach in 10 minutes?
  - **Media**: What media would you use?
  - **Quiz**: how would you quiz people?


The best way to have a good idea
is to have lots of ideas.
- Linus Pauling

Many of those ideas will be absolute crap!



That’s okay!
You gotta get through the bad ones to get to the good ones.

Brainstorm:
Domains you could teach

- Cooking
- Programming
- Languages
- How to dance to tik tok
- Yoga
- Fitness
- Music
- Identify poison ivy from other plants.
- GAMES
- Health
- Music theory
- Fashion
- Basketball rules – “what is a pick and roll”
- Dance
- Flowers
- Wine

Brainstorm:
Topics to teach in Cooking domain

- Is meat cooked?
- Learn different shapes
- How prepare sashimi
- How to fold dumplings!
- Vegan meat substitutes
- Mooncakes
- How to tell if fruit is fresh

Brainstorm:
Topics to teach in **Dance domain**

Tik tok dances

* every growing, memorize and do them

Moonwalk

Different grooves in hiphop

Fortnite dance

Stanky leg

- Specific song

Ballet positions

- 5 basic ones: explain what they are or how to add on to them.

Stretching for dance

Steps for ballroom dancing

What kind of dancing should I do in what setting.

Brainstorm:
Topics to teach in **Programming domain**

In heritance
- Client side: HTML JS, CSS
  - Stuff not taught in class
- How to debug
- GIT GITHUB
  - How to rebase
- ~~10 design principles~~
- MVC
- ~~Human-centered design~~
- ~~Infomration hierarchy~~
- ~~Affordances~~
- Latex / overleaf
  - How do I get stated?
  - Symbols and equations
  - How to place and image
  - TABLES

* What are types of list: make them look nice

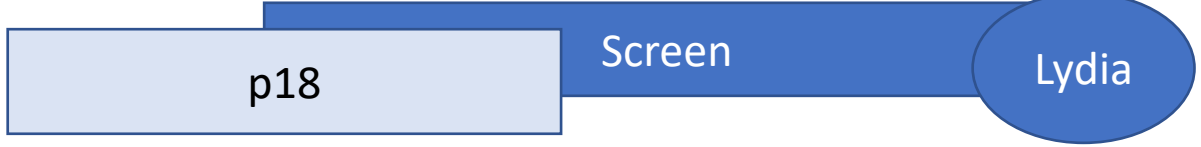
- How to do code
  - Software Engineering (Junfeng)

HTML basic elements

The difference between GET and POST ajax reuests

Warm-up (due Wednesday)
Group brainstorming domains and topics

- Meet your group and TA!
- Individual Brainstorm (5 minutes)
  - What are 5 domains that you could teach things to your classmates?
  - For each of those 5 domains, list 5 specific topics that you could teach interactively in under 10 minutes
- Group brainstorm:
  - Compare ideas, iterate!
  - **Users**: Would people in this class learn something from it?
  - **Topic**: is the topic focused enough to teach in 10 minutes?
  - **Media**: What media would you use?
  - **Quiz**: how would you quiz people?



p1

Empty row

p2

Empty row

p5

Empty row

p6

Empty row

p7

Empty row

p19

p9

Empty row

p10

Empty row

p11

Empty row

p13

Empty row

p15

Empty row

p16

Empty row

p17