
FRONT-END WEB DEVELOPMENT

SNACKS & DESIGN

TODAY!

KATRINA WANG, JASON LI

(GOOGLE SHEET IS PINNED IN SLACK)

FEWD

Q & A

“You mentioned that we're using IDs because we're doing javascript. I know you touched on that before, but I was wondering if you could explain that further? Why wouldn't classes work in the scenario?”

“How to loop through an array? (replace the first image with the last or vice-versa)”

“When working on our a site, is it better to build out the site using HTML and css, followed by adding jQuery / JS, or should jQuery/ JS be added in conjunction to building out the frameworks of a site? “

“Need practice working with loops”

“Do you have any real world applications of arrays that you can show us off the top of your head?”

Thursday March 24th - CSS Positioning & Animation

Sunday March 27th - Office Hours (*Adriana*)

Tuesday March 29th - Responsive Design

Thursday March 31st - Responsive + Final Project Lab (*Adriana*)

Sunday April 3rd - Office Hours (*Adriana*)

Tuesday April 5th - Guest Speaker: Mig Reyes! (plus drinks?)

Eric Boyer

SPROUT SOCIAL PANEL

Wed April 13th @ 6pm

(dinner provided)

CSS POSITIONING & ANIMATION

Eric Boyer

LEARNING OBJECTIVES

- Identify and differentiate between different CSS positioning techniques
- Familiarity with how animations and transitions can be used in CSS
- Understand how animation can still be controlled using JS
- Know the different ways to debug code and how to apply the concepts

AGENDA



- Github Branching
- Debugging
- Carousel Lab Pt. 2
- Advanced CSS Positioning
- Transitions
- Transformations
- Animations

CHIFEWD7

FINAL PROJECTS

Create a folder and push to Github

Discuss the idea of branching

<https://guides.github.com/introduction/flow/>

JQUERY

DEBUGGING



WHY ISN'T IT WORKING?

DEBUGGING — WHERE TO START

This will tell you where to start your hunt.



THE IMAGE IS NOT MOVING

*Find the code that makes
the image move*



NONE OF MY CODE WORKS

** Syntax error, check console*

DEBUGGING

To access debugging console:

PC: CTRL+SHIFT+J

Mac: COMMAND+OPTION+J

Click the error

DEBUGGING — LEVEL 1

Check for errors in console

- The location may not be correct but is a good place to start.
- Ex: Unbalanced brackets or parentheses



Uncaught SyntaxError: Unexpected token)

main.js:13



DEBUGGING — LEVEL 2

▸ `console.log()` can be used to display variable values in the console. This is useful for debugging.

```
console.log(variableName);
```

This should print the element to the console.
If it doesn't, there's probably something wrong with your selector.

**Shortcut to access console: cmd + opt + j*

DEBUGGING — LEVEL 3

Do some Googling!

- Try Googling it
- Be ready to clearly articulate the problem (Write out what your problem is)

Use Slack!

JQUERY

CAROUSEL LAB — PART 2

DECLARING ARRAYS

```
var descriptiveNameHere = [item1, item2, item3];
```


ARRAYS — ACCESSING ITEMS IN AN ARRAY

Accessing items in array:

`myArray[1]` → `true`

`myArray[2]` → `2`

`myArray[0]` → `5`

`myArray[3]` → `'Hello'`

```
var myArray = [5, true, 2, 'Hello']
```

ARRAYS – ADDING A VALUE/REPLACING A VALUE

INSERTING A NEW VALUE

- ▶ We can insert new values into any space in the array using the positions index.

```
myArray[1] = 'Hello';
```

UPDATING VALUES

- ▶ If there's already an item at that position, it will be replaced with the new value.

```
var myArr = [65, 'hello', true];  
myArr[1] = 'goodbye';  
// myArr[1] now holds 'goodbye' instead of 'hello'
```

ARRAYS - LENGTH

- We can use the `.length` property to find out how many items are in an array

```
var shapes = ['circle', 'triangle', 'square'];
```

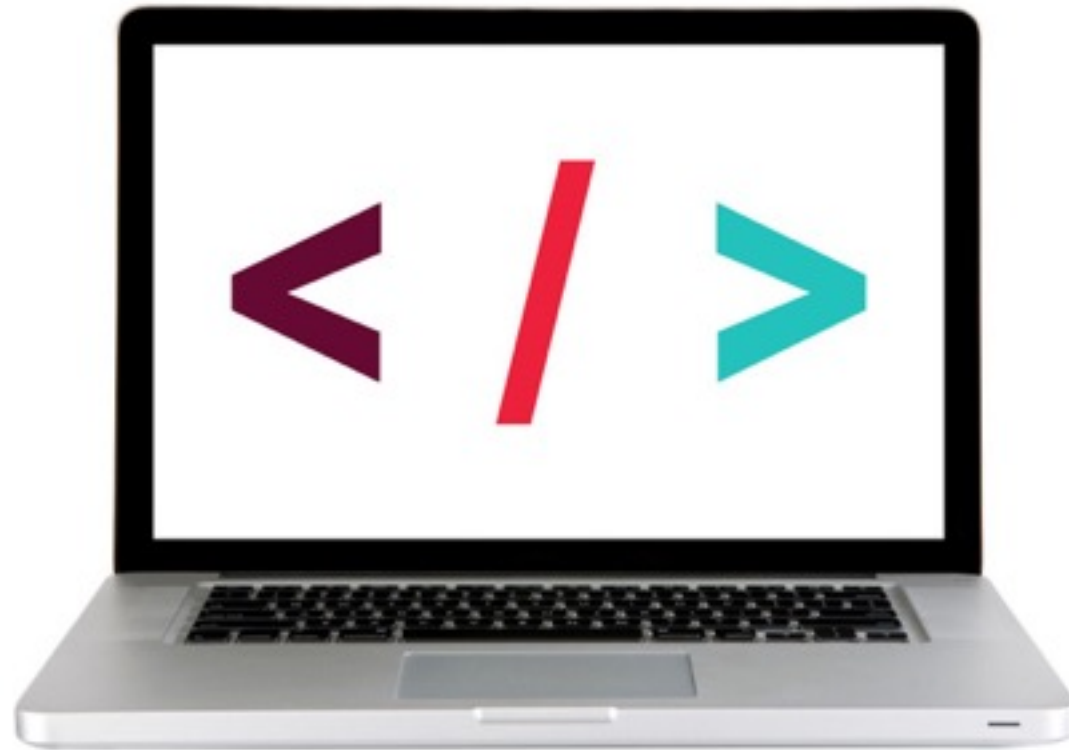
```
shapes.length;    => 3
```

- Accessing the last element in an array:

```
var myFavoriteShape = shapes[shapes.length-1];    => Prints 'square'
```

CAROUSEL WALK-THROUGH

CAROUSEL LAB



JQUERY

‘THIS’ KEYWORD

THE KEYWORD 'THIS'

this refers to whatever you *selected* with jQuery

```
$('p').on('click', function(){  
    $(this).fadeOut(500);  
});
```

JQUERY — SELECTING ELEMENTS

Selector



```
$('li').addClass('selected');
```


ACTIVITY



EXERCISE

KEY OBJECTIVE

- Practice applying the `this` keyword

TYPE OF EXERCISE

- Individual/Partner

TIMING

6 min

1. Follow the instructions in `lesson11_starter_code > [1]`
`this > js > main.js`

JQUERY

PREVENT DEFAULT

JQUERY METHODS — THE EVENT OBJECT

- ▶ The event object has properties and methods that tell you more about the event that took place.
- ▶ By using the preventDefault method, the default action of the event will not be triggered.

Parameter name

```
$('#li').on('eventGoesHere', function(e) {  
    e.preventDefault();  
});
```

Use that name in the function and use dot notation to access its properties and methods.

**CREATE
EVENT
LISTENERS**

CSS REVIEW

ADVANCED CSS POSITIONING REVIEW

STATIC POSITIONING

- This is the normal flow of the document, the **default**
- Elements render in order, as they appear in the document flow.

```
.my-class {  
  position: static;  
}
```

RELATIVE POSITIONING

- Relative positioning moves an element *relative to where it would have been in normal flow*.
- For example, "left: 20px" adds 20px to an element's **left** position
- Creates a *coordinate system* for child elements.

```
.my-class {  
  position: relative;  
  top: 20px;  
  left: 30%;  
}
```

ABSOLUTE POSITIONING

- When the *position* property is given a value of *absolute*, an element is taken out of the normal flow of the document.
- This element no longer affects the position of other elements on the page (they act like it's not there).
- You can add the *right*, *top*, *left* and *bottom* properties to specify where the element should appear relative to its first positioned (not static) ancestor element

```
.my-class {  
  position: absolute;  
  top: 0;  
  left: 500px;  
}
```

FIXED POSITIONING

- When the *position* property is given a value of *fixed*, the element is positioned in relation to *the browser window*
- When the user scrolls down the page, it stays in the same place.
- You can add the *right*, *top*, *left* and *bottom* properties to specify where the element should appear in relation to the browser window.

```
.my-class {  
  position: fixed;  
  top: 0;  
  left: 500px;  
}
```

OVERLAPPING ELEMENTS — Z-INDEX

- When using relative, fixed or absolute positioning, elements can overlap.
- When elements overlap, the elements that appear later in the HTML code sit on top of those that appear earlier in the page.
- If you want to control which elements are layered on top of each other, you can use the z-index property.
- This property takes a number — the higher the number the closer that element is to the front.
- Similar to 'bring to front' and 'send to back' in programs like *Adobe Illustrator*.

```
.my-class {  
  z-index: 10;  
}
```

WANT TO LEARN MORE?

Resources for more info/examples:

- ▶ A List Apart: [CSS Positioning 101](#)

ANIMATION

TRANSITIONS

TRANSITIONS

- Provide a way to control animation speed when changing properties
- Instead of having property changes take effect immediately, you can have them take place over a period of time.

```
.example {  
  transition: [transition-property] [transition-duration] [transition-timing-function] [transition-delay];  
}
```

TRANSITIONS



PROPERTY

Which properties
to animate



DURATION

How long the
transition will last



**TIMING
FUNCTION**

How the transition
will run



DELAY

When the animation
will start

TRANSITIONS – TRANSITION-PROPERTY

PROPERTIES
TO ANIMATE

- ▶ Can specify a specific property to transition or "all" to transition all properties
- ▶ *Default:* all

```
div {  
  transition: opacity 0.5s;  
}
```

```
div {  
  transition: all 0.5s;  
}
```

```
div {  
  transition: height 0.5s;  
}
```

```
.example {  
  transition: [transition-property] [transition-duration] [timing-function] [transition-delay];  
}
```

TRANSITIONS – TRANSITION-DURATION

DURATION

- ▶ A time value, defined in seconds or milliseconds

```
div {  
  transition: all 0.5s;  
}
```

```
div {  
  transition: all 350ms;  
}
```

```
div {  
  transition: all 3s;  
}
```

```
.example {  
  transition: [transition-property] [transition-duration] [timing-function] [transition-delay];  
}
```

TRANSITIONS

TIMING FUNCTION

- Describes how a transition will proceed over its duration, allowing a transition to change speed during its course.
- Timing functions: ease, linear, ease-in, ease-out, ease-in-out

```
div {  
  transition: opacity 0.5s ease;  
}
```

```
div {  
  transition: opacity 0.5s ease-in-out;  
}
```

```
.example {  
  transition: [transition-property] [transition-duration] [timing-function] [transition-delay];  
}
```

transition timing: [W3 Schools](https://www.w3schools.com/css/css3_transitions.asp)

TRANSITIONS



DELAY

- ▶ Length of time before the transition starts

```
div {  
  transition: background-color 0.5s ease 2s;  
}
```

```
.example {  
  transition: [transition-property] [transition-duration] [timing-function] [transition-delay];  
}
```

MORE FUN WITH TRANSITIONS — CODROPS

Fun CSS button styles: [Creative buttons](#)

Icon hover effects: [Icon Hover Effects](#)

Modal dialogue effects (advanced): [Dialogue Effects](#)

ANIMATION

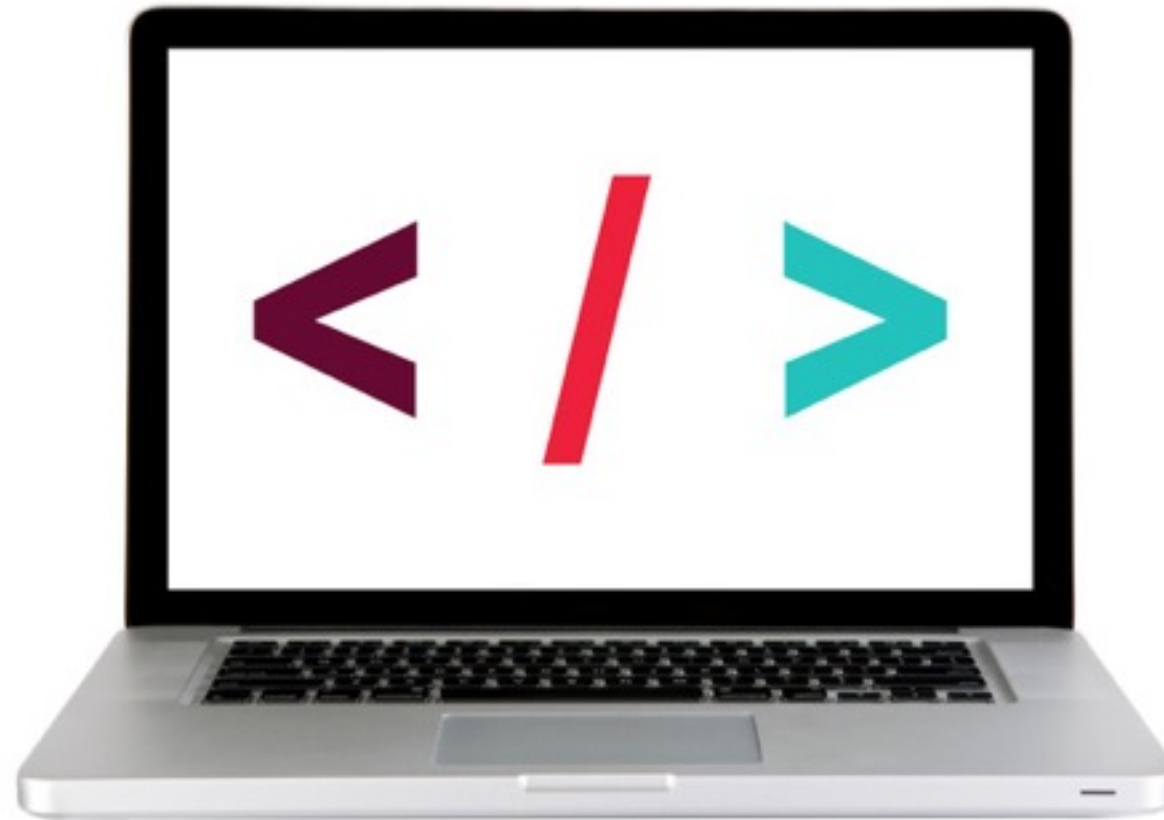
TRIGGERING TRANSITIONS

TRIGGERING TRANSITIONS

There are two ways to trigger CSS transitions:

1. Using the :hover CSS pseudo-class
2. Adding a class with jQuery

LET'S TAKE A CLOSER LOOK — TRIGGERING TRANSITIONS



ACTIVITY — BUTTON LAB



EXERCISE

KEY OBJECTIVE

- Practice using CSS transitions

TYPE OF EXERCISE

- Individual/Partner Lab

TIMING

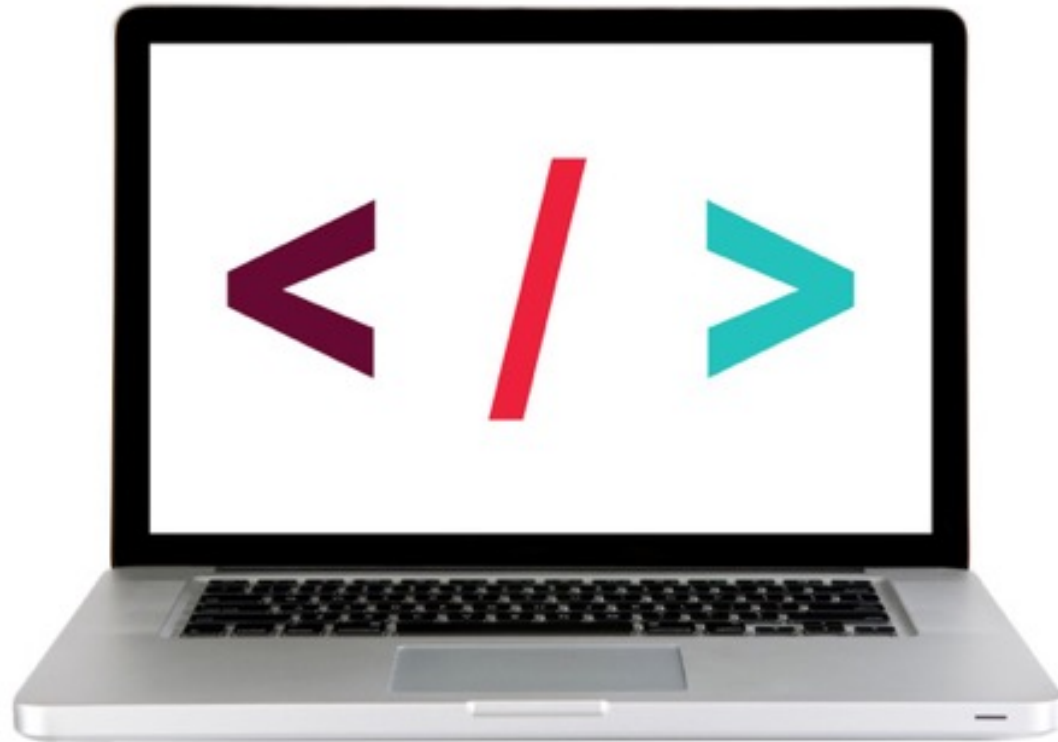
6 min

1. Add :hover styles and transition to the button:
lesson11_starter_code > [2] transition_button_lab

ANIMATION

TRANSFORMATIONS

LET'S TAKE A CLOSER LOOK — TRANSFORM



transform: [W3 Schools](#)
transform-origin: [W3 Schools](#)

ACTIVITY — HAMBURGER ICON



EXERCISE

KEY OBJECTIVE

- Practice using CSS transitions

TYPE OF EXERCISE

- Individual/Partner Lab

TIMING

10 min

1. Follow the instructions in `lesson11_starter_code > [3] transformation_lab`

ANIMATION

ANIMATIONS

KEYFRAME ANIMATIONS

- ▶ Keyframe animations allow developers to create smooth, maintainable animations that perform well and don't require tons of scripting



KEYFRAME ANIMATIONS — SYNTAX

1. Define custom animation

```
@-webkit-keyframes NAME-YOUR-ANIMATION {  
  0%   { opacity: 0; }  
  100% { opacity: 1; }  
}  
@keyframes NAME-YOUR-ANIMATION {  
  0%   { opacity: 0; }  
  100% { opacity: 1; }  
}
```

2. Assign using the animation property

```
#box {  
  -webkit-animation: NAME-YOUR-ANIMATION 5s infinite;  
  animation:        NAME-YOUR-ANIMATION 5s infinite;  
}
```

A simple tool to make sure you're including all the necessary browser prefixes: [pleasease](#)

KEYFRAME ANIMATIONS — ANIMATION PROPERTY

Properties:

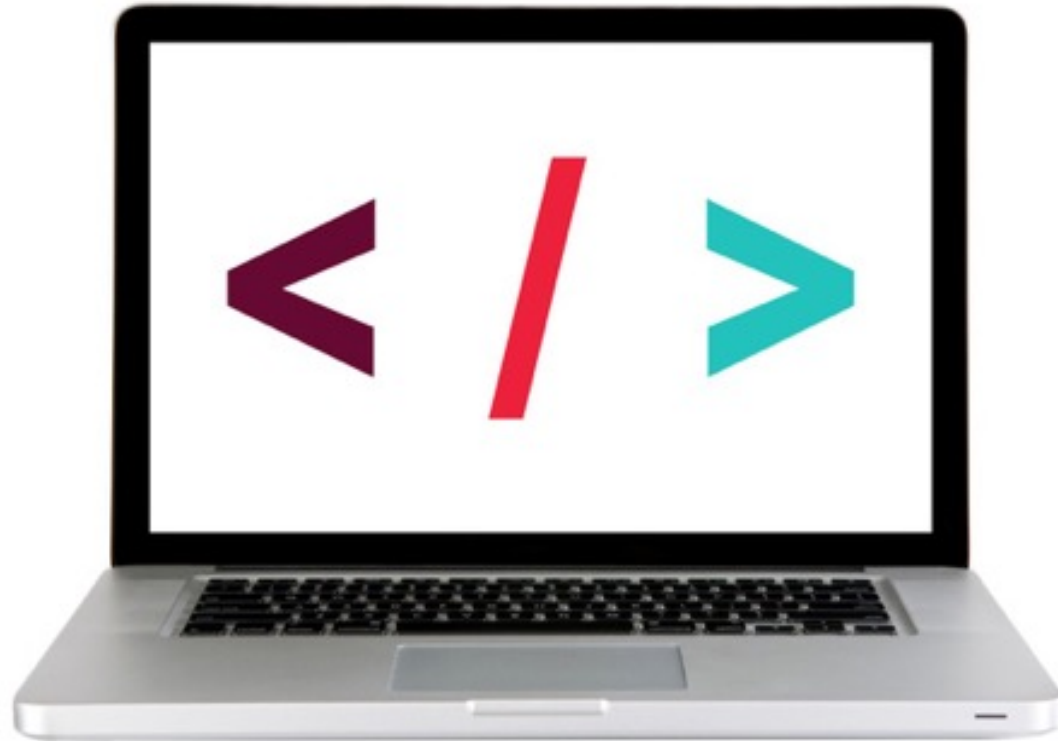
```
.box {  
  animation-name: bounce;  
  animation-duration: 4s;  
  animation-timing-function: ease-out;  
  animation-delay: 2s;  
  animation-iteration-count: 10;  
}
```

Shorthand:

```
.example {  
  animation: name duration timing-function delay iteration-count;  
}
```

```
.example {  
  animation: bounce 4s ease-out 2s 10;  
}
```

LET'S TAKE A CLOSER LOOK



Code along — Spinning Wheel

ANIMATION

LAB

LAB



ACTIVITY



EXERCISE

KEY OBJECTIVE

- Practice triggering CSS transitions with jQuery

TYPE OF EXERCISE

- Individual/Partner Lab

TIMING

10 min

Until 8:50

1. Test the live demo (ERIC, DEPLOY THIS)
2. Add base styles to the page **lesson11_starter_code > [5] sidebar_lab**
3. Make sidebar interactive using jQuery and CSS transitions.

LEARNING OBJECTIVES

- Identify and differentiate between different CSS positioning techniques
- Familiarity with how animations and transitions can be used in CSS
- Understand how animation can still be controlled using JS
- Know the different ways to debug code and how to apply the concepts

LAB

HOMEWORK

HOMEWORK

FEEDBACK GROUPS FOR WIREFRAMES

FEEDBACK GROUPS

IF YOU ARE STUCK...

- 1) Use the “Chrome Inspector” to look at your code, use the console, Google for answers.
 - 2) Ask your question in the Slack channel and see if any fellow students might know the answer
 - 3) Ask Eric & Adriana
- * When using your fellow students and instructors, pushing your code to Github is a great place to share where all of your code is currently at.

FEEDBACK GROUPS

GROUP 1 (ERIC)

Katie Allyn
Kimberly Baird
Tom Bunting
Michael Fischer
Jim Howes
Jon Iler
Peyton Lee
Anna Matras
Christopher Zalek

TO: EBOYER@GMAIL.COM

GROUP 2 (ADRIANA)

Corin Menuge
Colin Quirk
Allison Schaffer
Kevin Sella
Michael Sena
Nicholas Skeba
Ashley Sullivan
Lisa Vasquez
Katrina Wang
Xi Yue Li

TO: ADRIANA.LCS316@GMAIL.COM

FEEDBACK GROUPS

DUE 11:59PM, SUNDAY MAR 13TH

- Complete Final Project Wireframes
- Optionally: finish today's lab, tuesdays lab, or a variation of any of them.
- Add files, Commit, and Push/Publish your homework to your class Github Repo.
(For Wireframes this can be a picture of a drawing)

FINAL PROJECT MILESTONE 2: PROPOSAL CHECK-INS W/ ERIC

~~Katie Allyn~~

~~Kimberly Baird~~

~~Tom Bunting~~

~~Michael Fischer~~

Jim Howes

~~Jon Iler~~

Peyton Lee

~~Anna Matras~~

~~Christopher Zalek~~

Corin Menuge

Colin Quirk

Allison Schaffer

~~Kevin Sella~~

~~Michael Sena~~

~~Nicholas Skeba~~

Ashley Sullivan

Lisa Vasquez

Katrina Wang

Jason Li

FRONT-END WEB DEVELOPMENT

SNACKS & DESIGN

TUESDAY MAR 29TH

ERIC & ADRIANA

(GOOGLE SHEET IS PINNED IN SLACK)

EXIT TICKETS