can floating affect a position

Yes! You can’t float objects you are using the position property on

I would like to know more about assigning widths to elements

If an element is display inline, you can’t assign a width

Can we align objects relative to another object? For example, align the bottom of nav text with the bottom of the header text.

You can but not the way you’re probably hoping. You need to play around with css properties like margin and padding.

How to use float and clearfix properly on specific elements.

Let’s go over together

what kinds of elements/classes can you use absolute/relative position on?

Anything you want!
Just make sure to put position relative on parent element and position absolute on the child element

AGENDA

- Final Project Overview
- Review
- Planning 101
- Intro to Programming
- Intro to Pseudo Code
- Intro to JS
- Reading JS
- Lab

FINAL PROJECTS

FINAL PROJECTS

WHERE CAN I GET SOME INSPIRATION FROM WHAT PAST STUDENTS HAVE DONE?

- Visit the General Assembly Gallery

WHERE SHOULD I BE RIGHT NOW?

- I am going to meet with you to talk ideas with you!
- Once your idea is approved, think about how you want it to look

REVIEW

REFACTORING

WHAT IS REFACTORING?

- Code refactoring is the process of restructuring existing computer code—changing the factoring—without changing its external behavior
- Refactoring improves nonfunctional attributes of the software

REFACTORING

Write D.R.Y. Code!!!
(Don’t Repeat Yourself!)

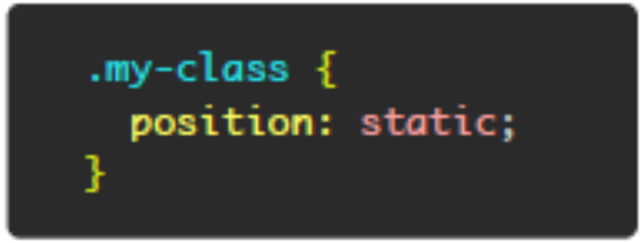
Code should be easy to read!

What is easy to read?

- Indented Code
- Helpful comments
- Organized
- Clear naming

STATIC POSITIONING

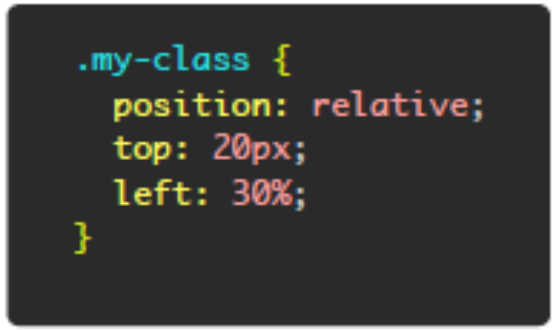
- This is the normal flow of the document, the **default**
- Elements render in order, as they appear in the document flow.



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

RELATIVE POSITIONING

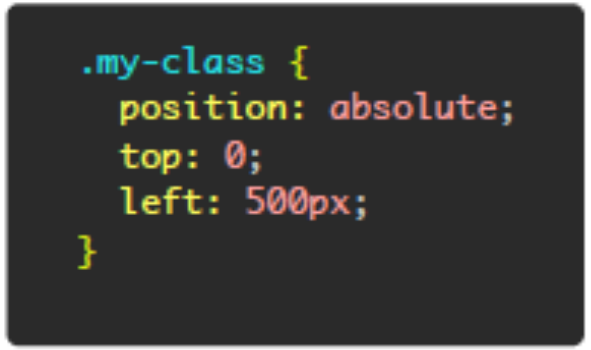
- Relative positioning moves an element *relative to where it would have been in normal flow.*
- For example, "left: 20px" adds 20px to an element's **left** position
- **Creates a coordinate system for child elements.**



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

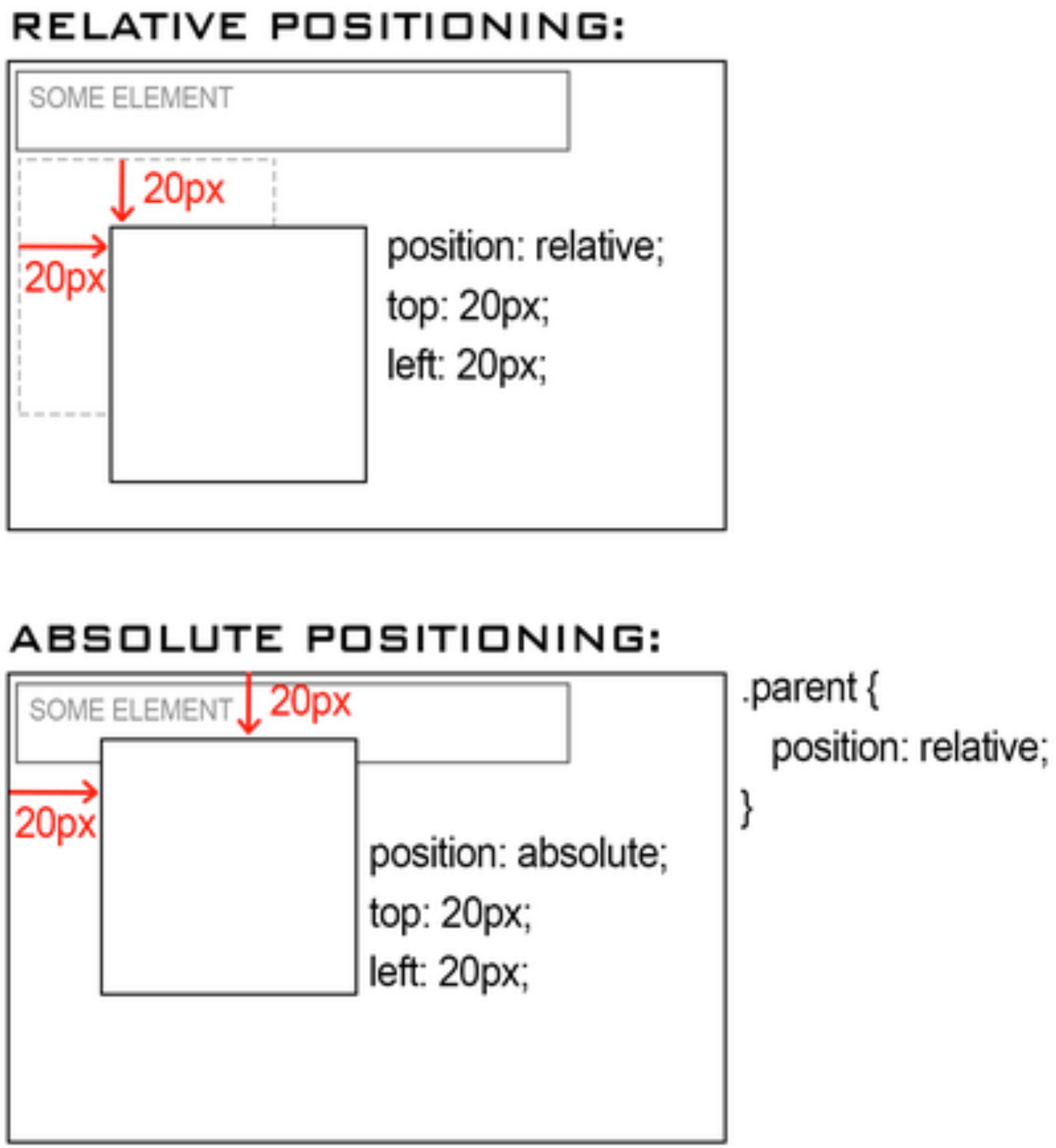
ABSOLUTE POSITIONING

- 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;
}
```

RELATIVE VS ABSOLUTE POSITIONING



RELATIVE POSITIONING:

```
position: relative;
top: 20px;
left: 20px;
```

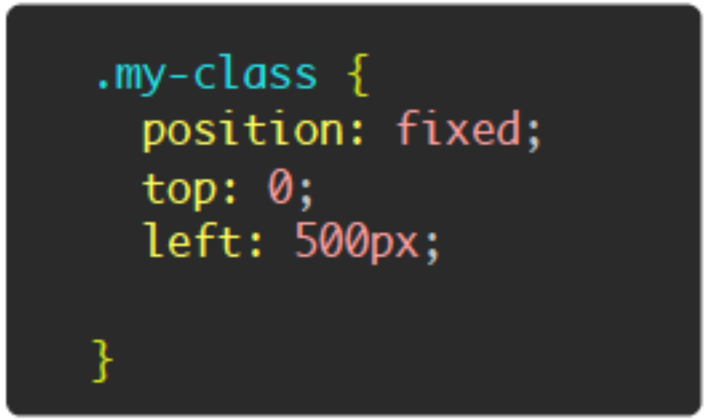
ABSOLUTE POSITIONING:

```
position: absolute;
top: 20px;
left: 20px;
```

```
.parent {
  position: relative;
}
```

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.

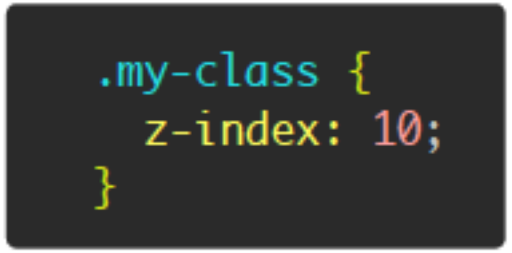


```
.my-class {
  position: fixed;
  top: 0;
  left: 500px;

}
```

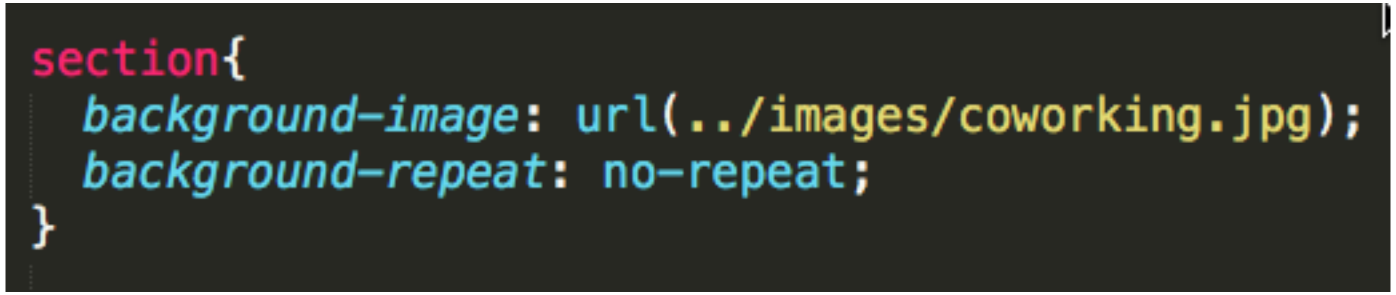
OVERLAPPING ELEMENTS — Z-INDEX

- 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*.

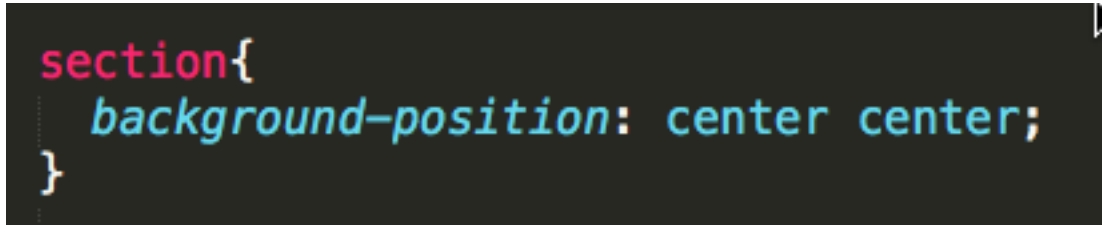


```
.my-class {
  z-index: 10;
}
```

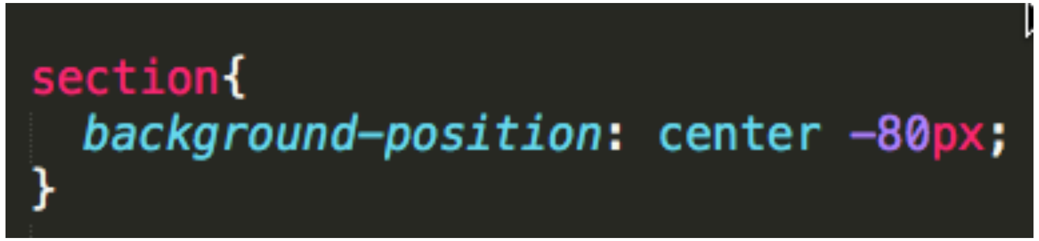
BACKGROUND PROPERTY



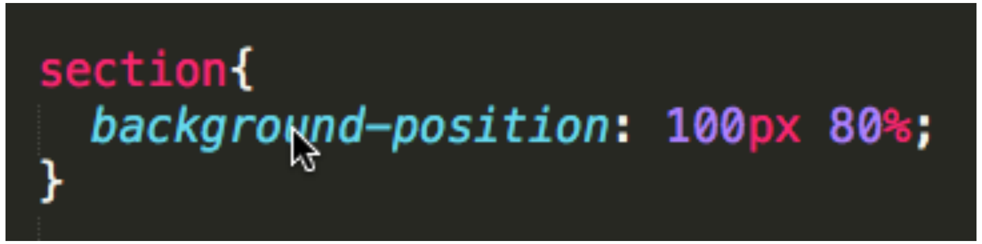
```
section{
  background-image: url(../images/coworking.jpg);
  background-repeat: no-repeat;
}
```



```
section{
  background-position: center center;
}
```



```
section{
  background-position: center -80px;
}
```



```
section{
  background-position: 100px 80%;
}
```

PLANNING 101 + PROGRAMMING INTRO

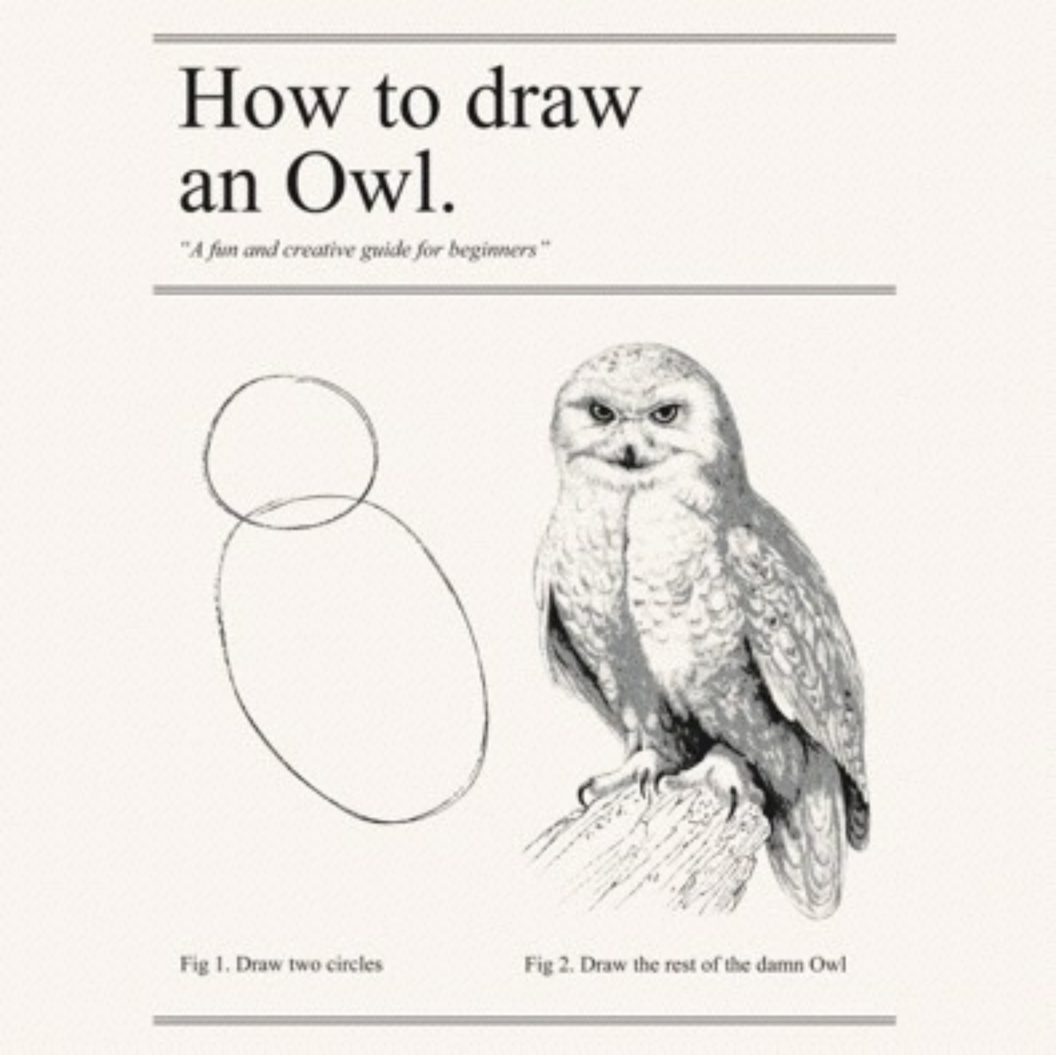
LESSON 6

LEARNING OBJECTIVES

- Learn basic planning tips
- Practice programmatic thinking by writing pseudo code to solve a basic problem.
- Define web site behavior and the practical uses of JavaScript.
- Predict DOM output / changes by reading JS code.

PLANNING 101

PLANNING



How to draw an Owl.

"A fun and creative guide for beginners"

Fig 1. Draw two circles

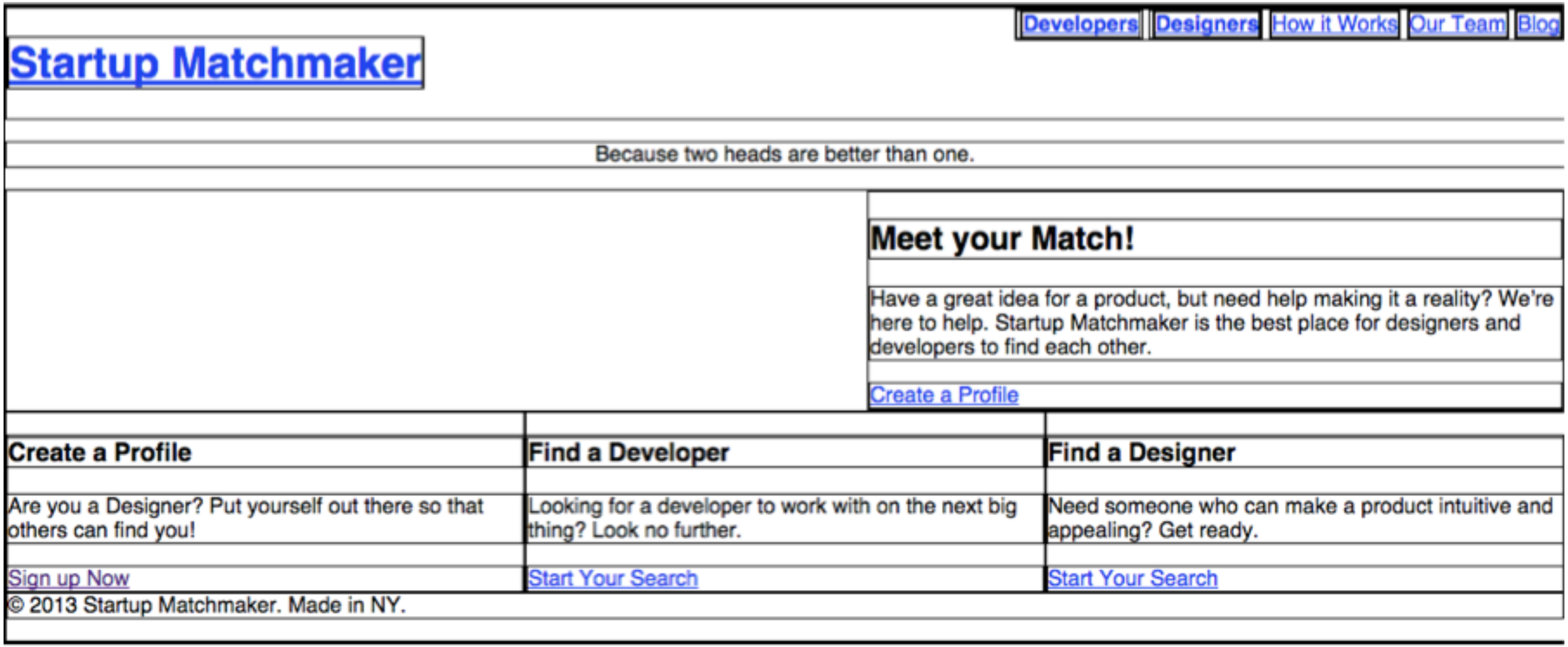
Fig 2. Draw the rest of the damn Owl

PLANNING 101

Le Steps

1. Write your HTML (use HTML5 Flowchart for guidance in picking elements)
2. Get things into place. (Add floats, clear: both, get columns set up, etc.)
3. Add base styles. (Font-family/color for the body, remove text-decoration under anchors and add a base anchor color, remove bullets for list items, etc.)
4. Work through the page section by section and start "filling in the details." Resist he urge to be a perfectionist at this point.
5. Polish things up! Compare the design with your page. Are you using the right fonts? Colors? Is any of the spacing off?

PLANNING 101



Developers | Designers | How it Works | Our Team | Blog

Startup Matchmaker

Because two heads are better than one.

Meet your Match!

Have a great idea for a product, but need help making it a reality? We're here to help. Startup Matchmaker is the best place for designers and developers to find each other.

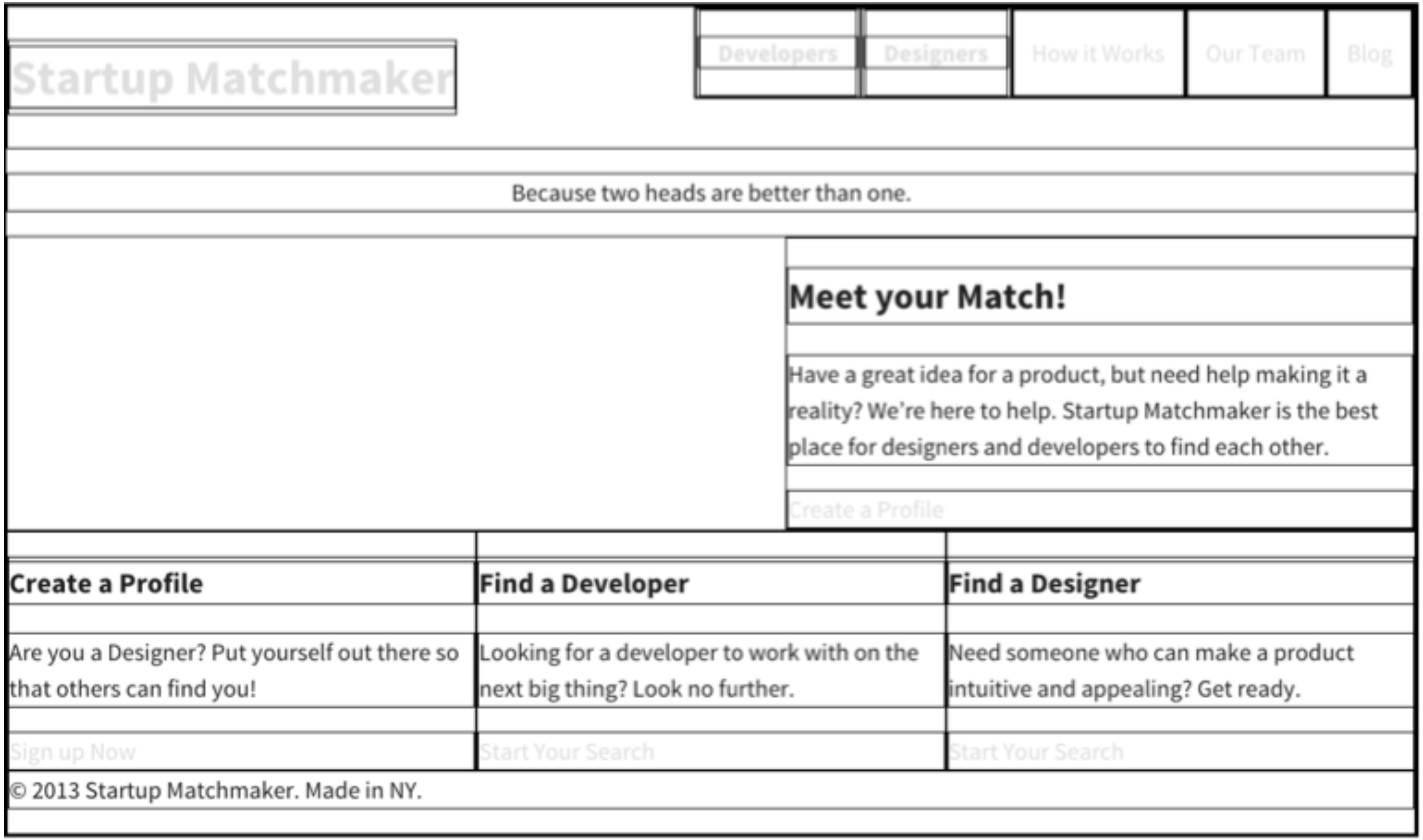
Create a Profile

Create a Profile	Find a Developer	Find a Designer
Are you a Designer? Put yourself out there so that others can find you!	Looking for a developer to work with on the next big thing? Look no further.	Need someone who can make a product intuitive and appealing? Get ready.
Sign up Now	Start Your Search	Start Your Search

© 2013 Startup Matchmaker. Made in NY.

*Pro tip: adding a border to everything on the page can help during this process: * {border: 1px solid black; }

PLANNING 101



Startup Matchmaker

Developers

Designers

How it Works

Our Team

Blog

Because two heads are better than one.

Meet your Match!

Have a great idea for a product, but need help making it a reality? We're here to help. Startup Matchmaker is the best place for designers and developers to find each other.

Create a Profile

Create a Profile

Are you a Designer? Put yourself out there so that others can find you!

Sign up Now

Find a Developer

Looking for a developer to work with on the next big thing? Look no further.

Start Your Search

Find a Designer

Need someone who can make a product intuitive and appealing? Get ready.

Start Your Search

© 2013 Startup Matchmaker. Made in NY.

INTRO TO PROGRAMMING

PROGRAMMING

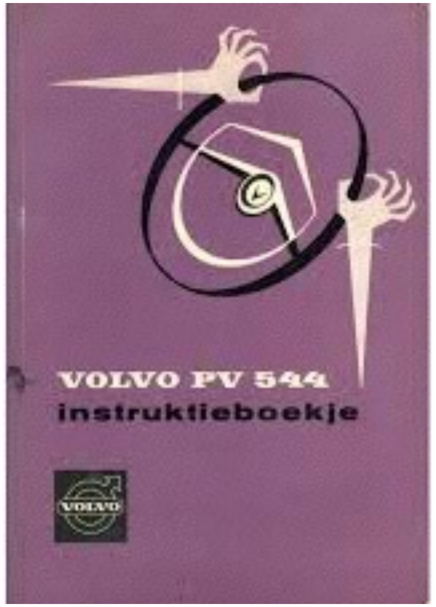
WHAT IS A PROGRAM?

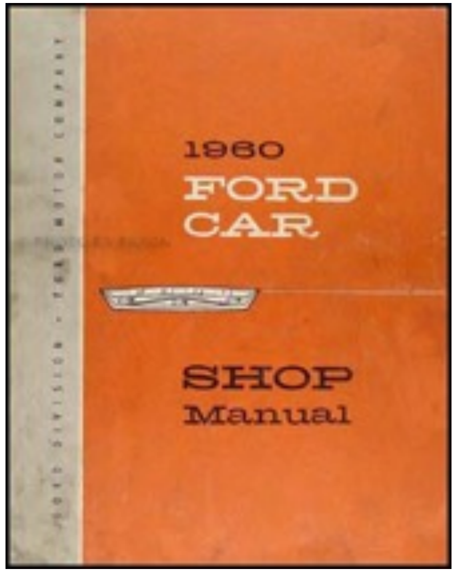
- A program is a set of instructions that you write to tell a computer what to do

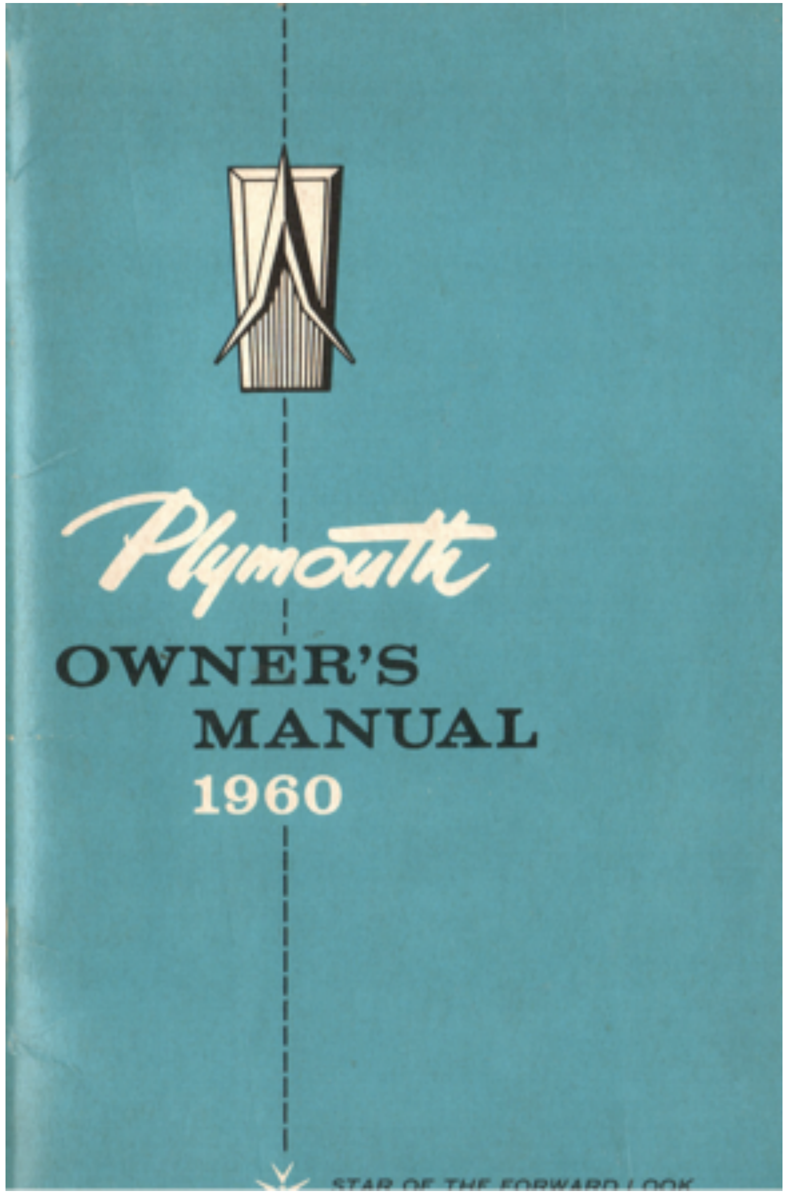
WHAT IS PROGRAMMING?

- Programming is the task of writing those instructions in a language that the computer can understand.

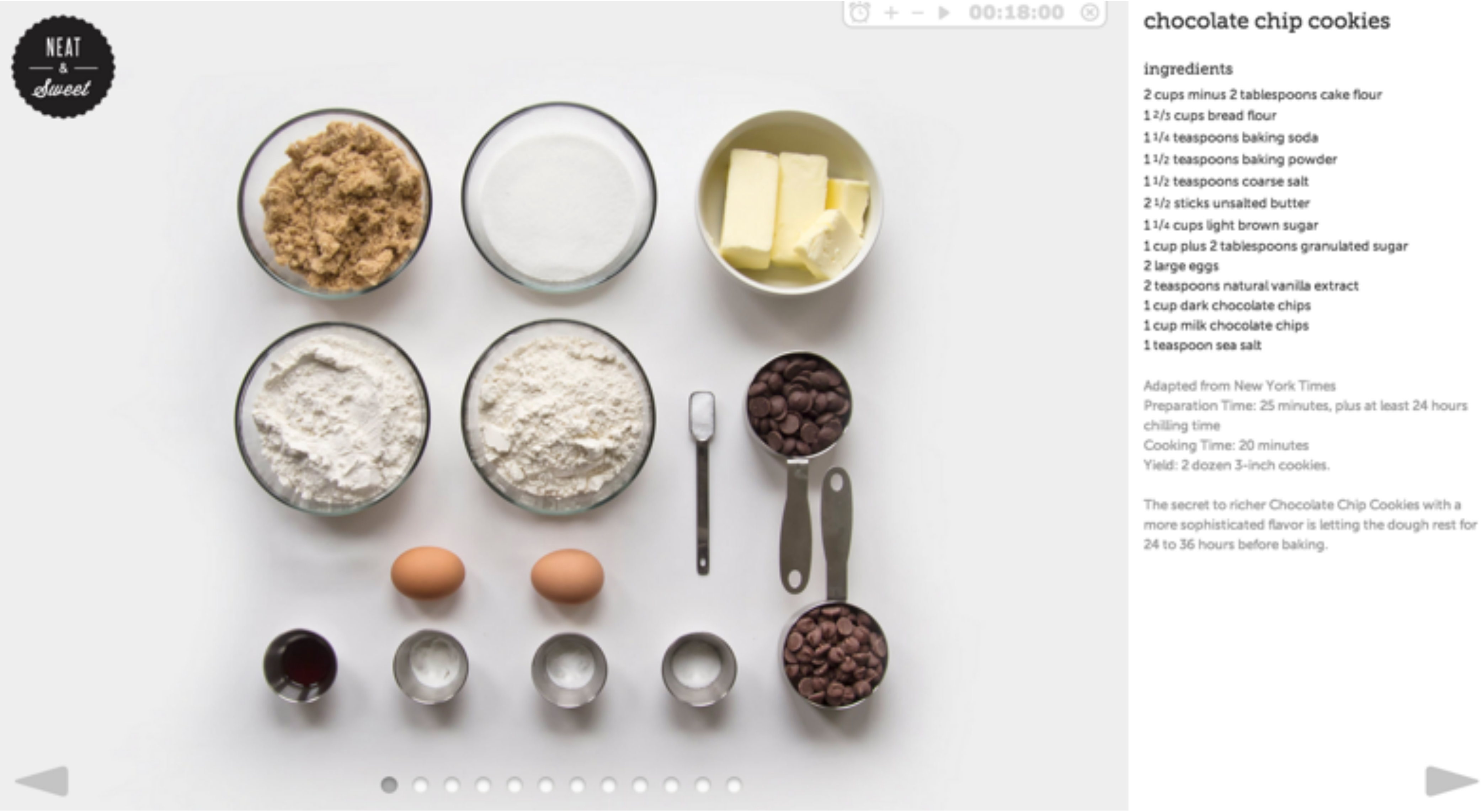
WHAT IS A PROGRAM?







WHAT IS A PROGRAM?



chocolate chip cookies

ingredients

2 cups minus 2 tablespoons cake flour
1 2/3 cups bread flour
1 1/4 teaspoons baking soda
1 1/2 teaspoons baking powder
1 1/2 teaspoons coarse salt
2 1/2 sticks unsalted butter
1 1/4 cups light brown sugar
1 cup plus 2 tablespoons granulated sugar
2 large eggs
2 teaspoons natural vanilla extract
1 cup dark chocolate chips
1 cup milk chocolate chips
1 teaspoon sea salt

Adapted from New York Times
Preparation Time: 25 minutes, plus at least 24 hours chilling time
Cooking Time: 20 minutes
Yield: 2 dozen 3-inch cookies.

The secret to richer Chocolate Chip Cookies with a more sophisticated flavor is letting the dough rest for 24 to 36 hours before baking.

HOW COMPUTERS “THINK”

- Short answer — they don’t think!
- While computers don’t think, they *act as if they do*, by sequentially executing simple instructions.
- The only things a computer knows are the things we tell it.
- A computer doesn't learn to perform tasks like you and I — it needs to follow instructions every time it performs the task.

INTRO TO PSEUDO CODE

PSEUDO CODE

- When we write a program, we need to figure out a way to translate the ideas that are in our heads into code

- Pseudo code is a way to 'plan out' your program before coding it
- **Pseudo code** is a detailed yet readable description of what a computer program must do, expressed in plain english rather than in a programming language

THE IMPORTANCE OF PLANNING

PSEUDO CODE — THERMOSTAT

Goal: *Write pseudo code for an application that would monitor the room temperature and adjust it so the room remains at a certain temperature*

CODE ALONG

PSEUDO CODE — ROCK PAPER SCISSORS

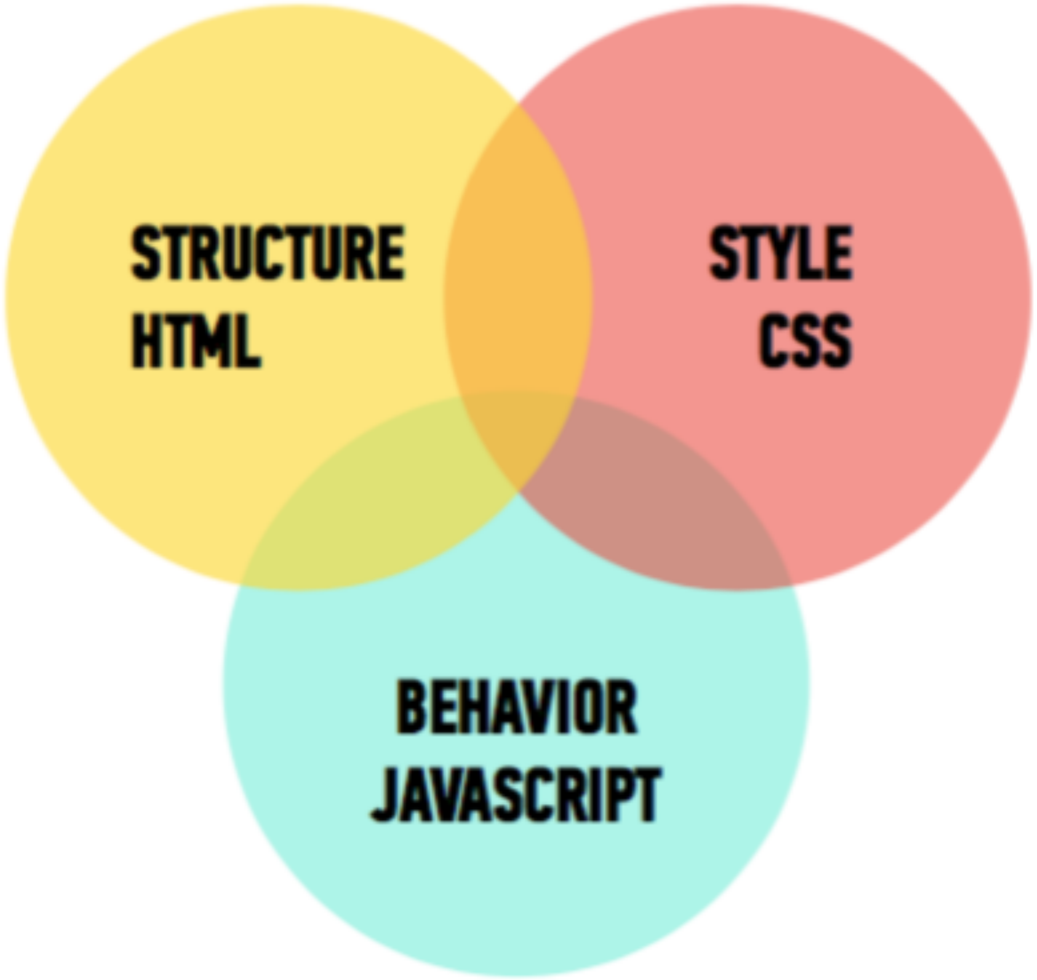
ROCK PAPER SCISSORS — 20 MIN

1. Get in groups of 3-4
2. Discuss how you would approach the problem
3. Write the steps in marker on the table
4. One person present how the group solved it

INTRO TO JS

THE THREE AMIGOS: STRUCTURE, STYLE, BEHAVIOR

- HTML = Noun
- CSS = Adjective
- Javascript = Verb



JAVA VS JAVASCRIPT

Just a quick note! We're learning **JavaScript** *in this class, not Java. Java and JavaScript are actually two different languages.*

!==

WHAT JAVASCRIPT CAN DO!

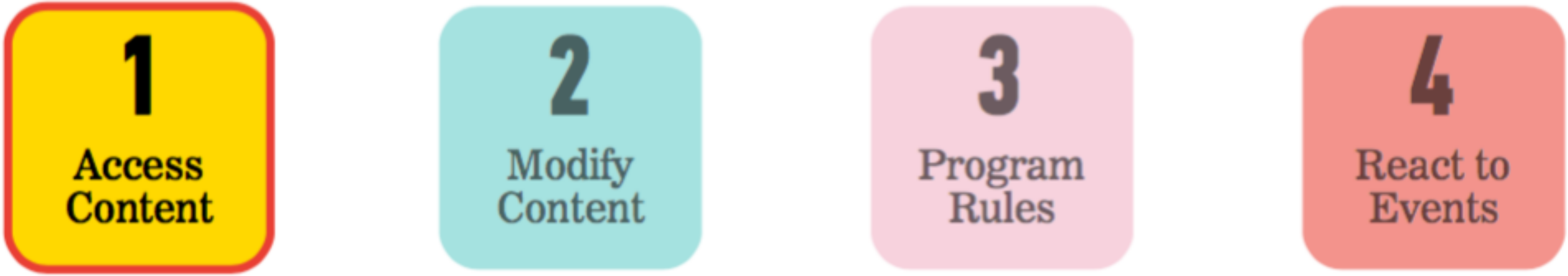
1 Access Content

2 Modify Content

3 Program Rules

4 React to Events

WHAT JAVASCRIPT CAN DO!



1 Access Content

2 Modify Content

3 Program Rules

4 React to Events

You can use JS to select any element, attribute or text from an HTML page.

For example:

- Select the text inside all the <p> elements on a page
- Select the element that has the id attribute with a value of **email**
- Find out what the user entered into a text input when they submit a form

WHAT JAVASCRIPT CAN DO!

1 Access Content

2 Modify Content

3 Program Rules

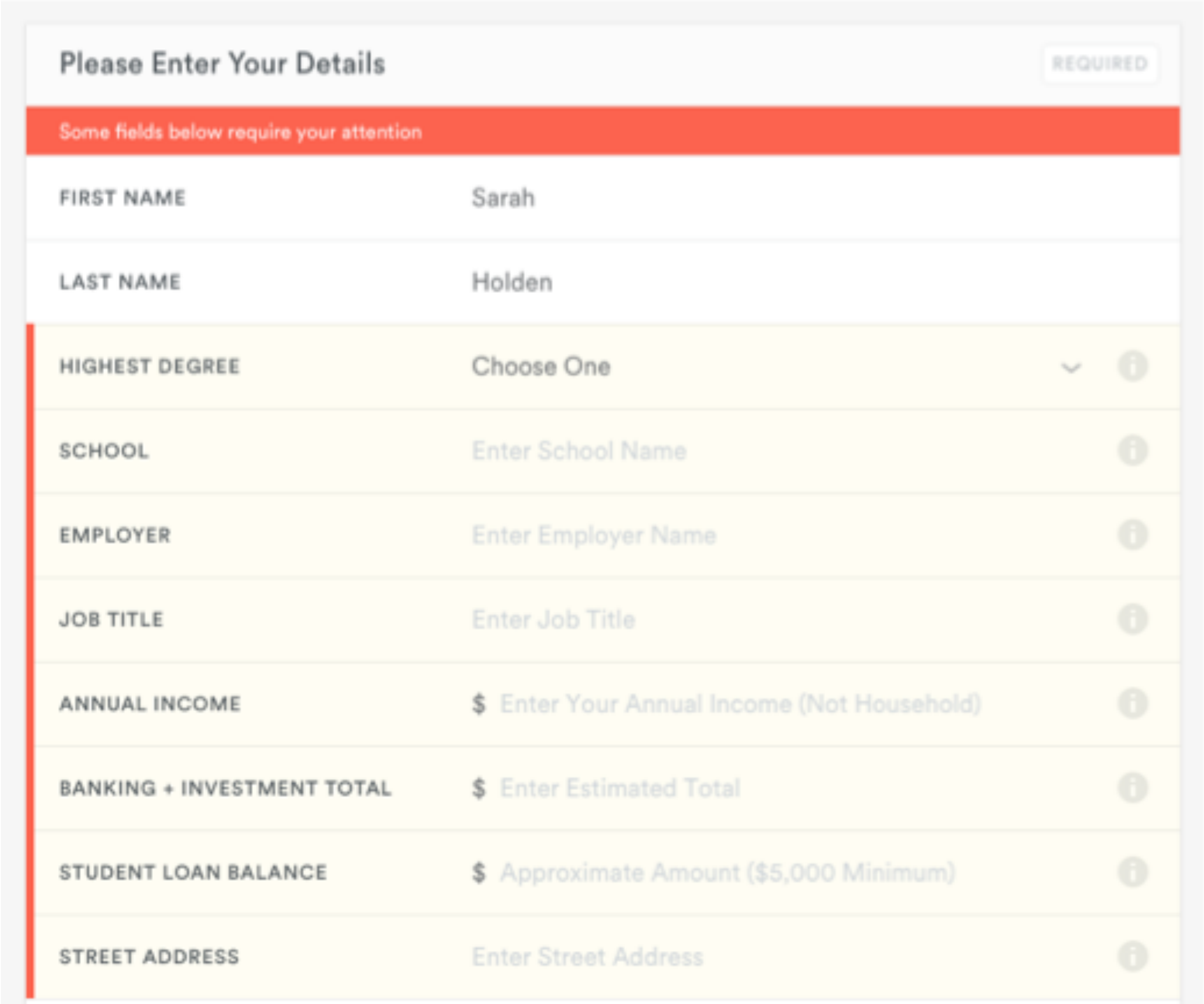
4 React to Events

You can use JS to add elements, attributes and text to the page (or remove them)

For example:

- Add an error message below a form
- Change the size, position, color, or other styles for an element
- Add or remove a class from elements to trigger new CSS rules for those elements

WHAT JAVASCRIPT CAN DO! — MODIFYING CONTENT



Please Enter Your Details

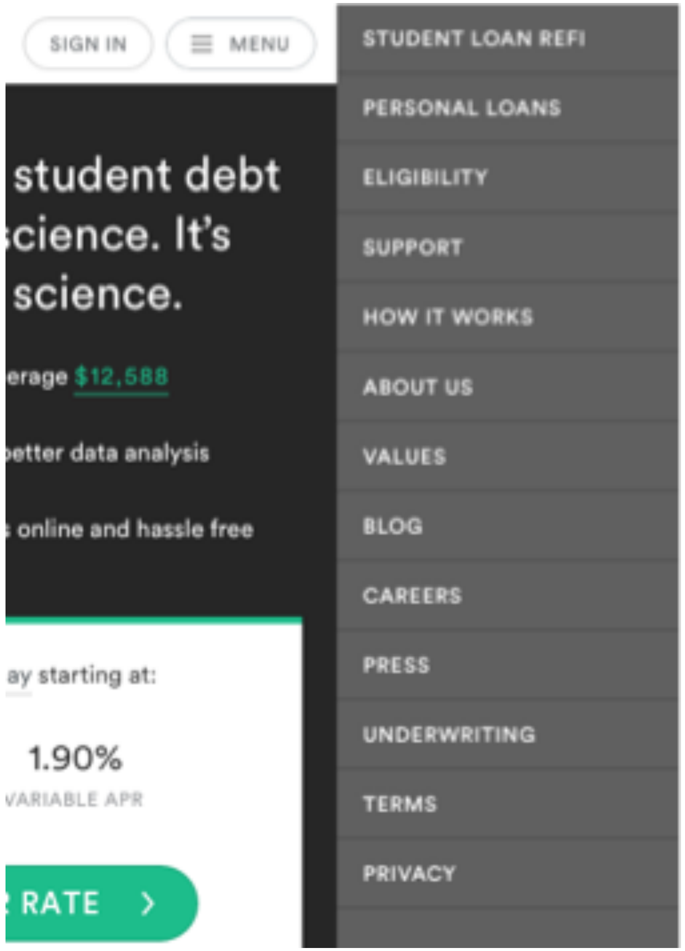
REQUIRED

Some fields below require your attention

FIRST NAME	Sarah
LAST NAME	Holden
HIGHEST DEGREE	Choose One
SCHOOL	Enter School Name
EMPLOYER	Enter Employer Name
JOB TITLE	Enter Job Title
ANNUAL INCOME	$ Enter Your Annual Income (Not Household)
BANKING + INVESTMENT TOTAL	$ Enter Estimated Total
STUDENT LOAN BALANCE	$ Approximate Amount ($5,000 Minimum)
STREET ADDRESS	Enter Street Address

Add an error message (and styles) to a form

WHAT JAVASCRIPT CAN DO! — MODIFYING CONTENT



student debt
cience. It's
science.

erage $12,588

etter data analysis

online and hassle free

ay starting at:

1.90%
VARIABLE APR

Change the size, position, color, or other styles for an element

WHAT JAVASCRIPT CAN DO!

1 Access Content

2 Modify Content

3 Program Rules

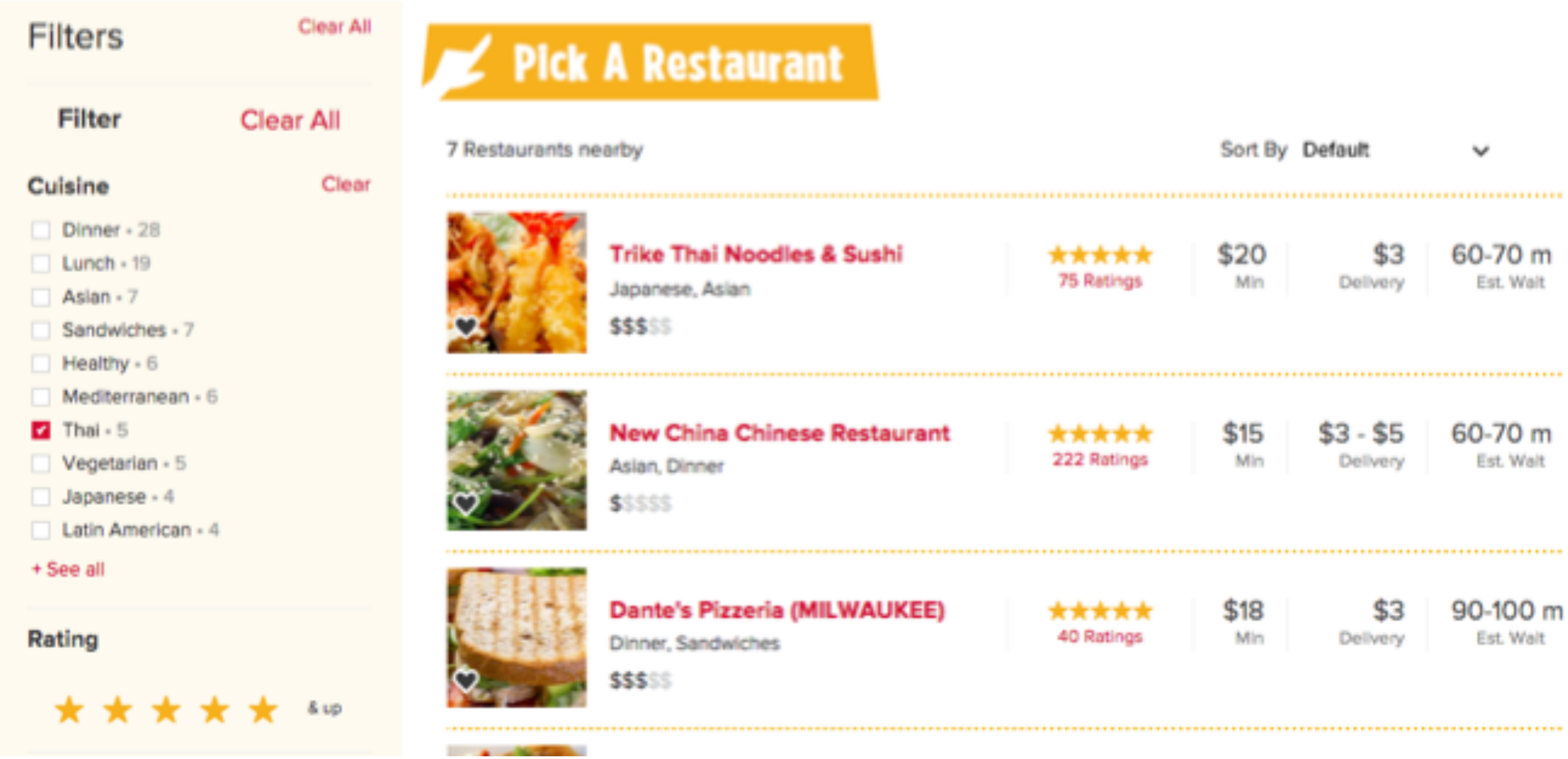
4 React to Events

You can specify a set of steps (instructions) for the browser to follow.

For example:

- Have images/text fade in as the user scrolls down the page
- Check to make sure the user has entered a valid email address into a form and display an error message if not
- Open a chat panel when the user clicks on a 'Chat with Us' button
- Filter data when the user selects a filter

WHAT JAVASCRIPT CAN DO! — PROGRAM RULES



Filters Clear All

Filter Clear All

Cuisine Clear

□ Dinner • 28
□ Lunch • 19
□ Asian • 7
□ Sandwiches • 7
□ Healthy • 6
□ Mediterranean • 6
☑ Thai • 5
□ Vegetarian • 5
□ Japanese • 4
□ Latin American • 4

+ See all

Rating

★ ★ ★ ★ ★ & up

Pick A Restaurant

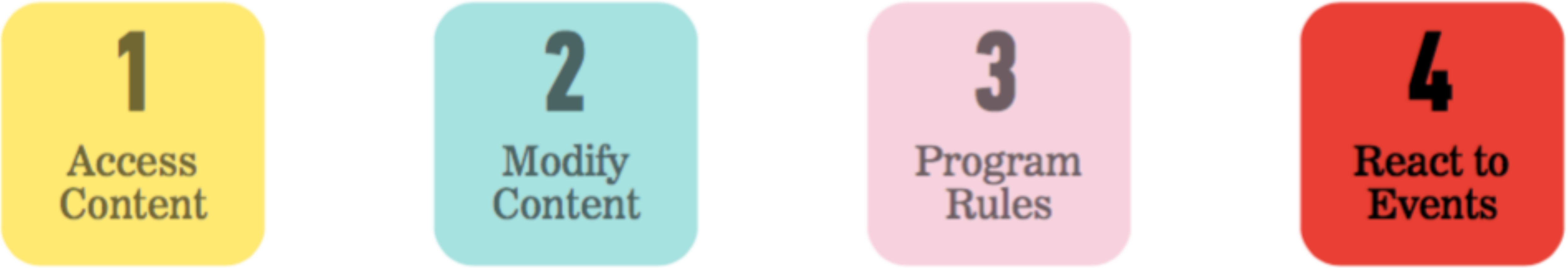
7 Restaurants nearby

Sort By Default

Trike Thai Noodles & Sushi Japanese, Asian $$$$$	★★★★★ 75 Ratings	$20 Min	$3 Delivery	60-70 m Est. Wait
New China Chinese Restaurant Asian, Dinner $$$$$	★★★★★ 222 Ratings	$15 Min	$3 - $5 Delivery	60-70 m Est. Wait
Dante's Pizzeria (MILWAUKEE) Dinner, Sandwiches $$$$$	★★★★★ 40 Ratings	$18 Min	$3 Delivery	90-100 m Est. Wait

Filter data when the user selects a filter

WHAT JAVASCRIPT CAN DO!



1 Access Content

2 Modify Content

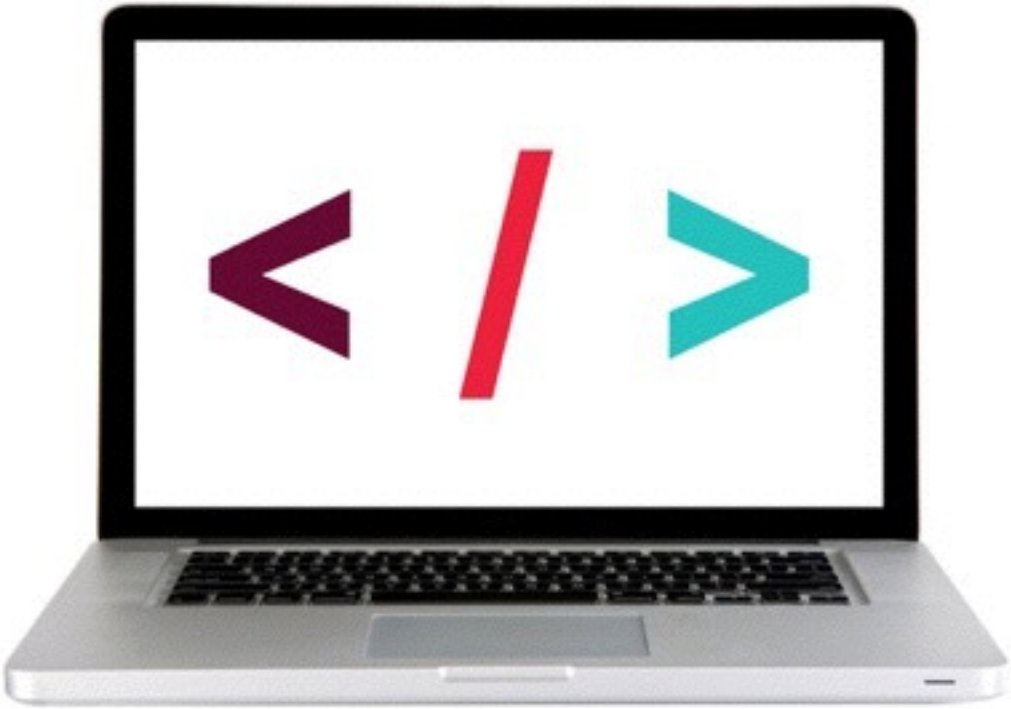
3 Program Rules

4 React to Events

You can specify that a script should run when an event occurs

For example:

- When a button is clicked
- When the cursor hovers over an element
- When the user types information into a form ‣ When a page has finished loading
- When the user hits enter to submit a form



HTTPS://KINHR.COM/PRICING/

READING JS

READING JS

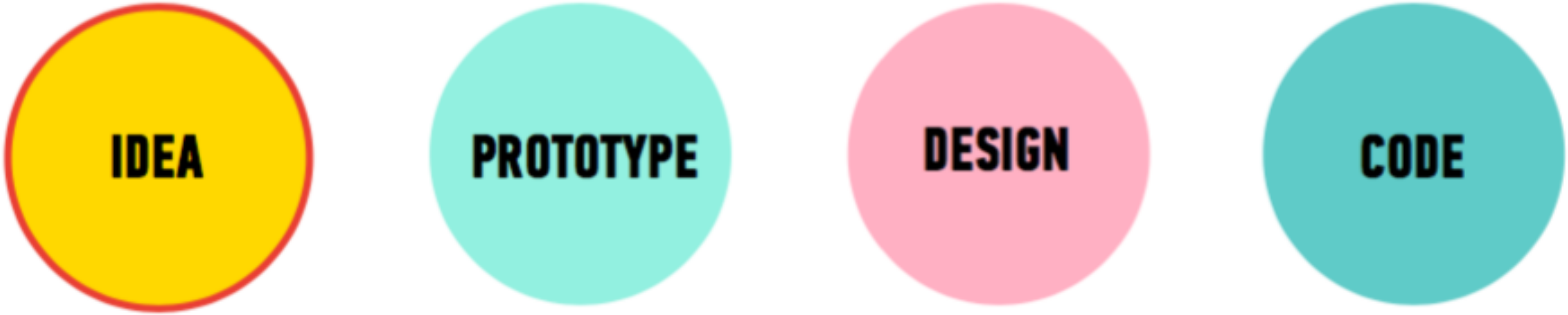
- When you are a child you learn to speak and read before you learn to write
- Developers learned to 'speak' JS with the discussion, video, and pseudo code

ACTIVITY

COLOR SWITCHER

WIREFRAMES

GETTING STARTED



IDEA

PROTOTYPE

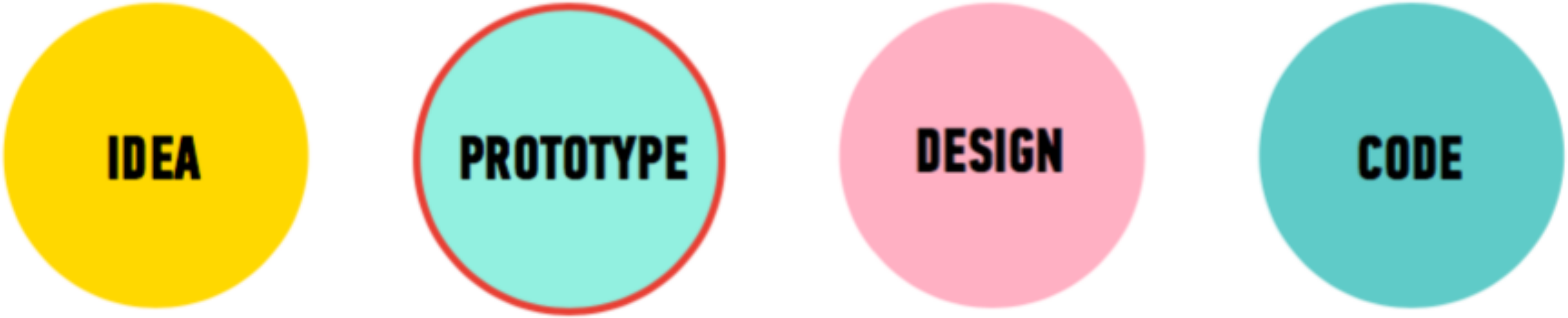
DESIGN

CODE

FIND A COMPELLING PROBLEM:

- What problem are you solving? And for whom?
- How is your "customer" solving this problem now?
- Why is your proposed solution better?

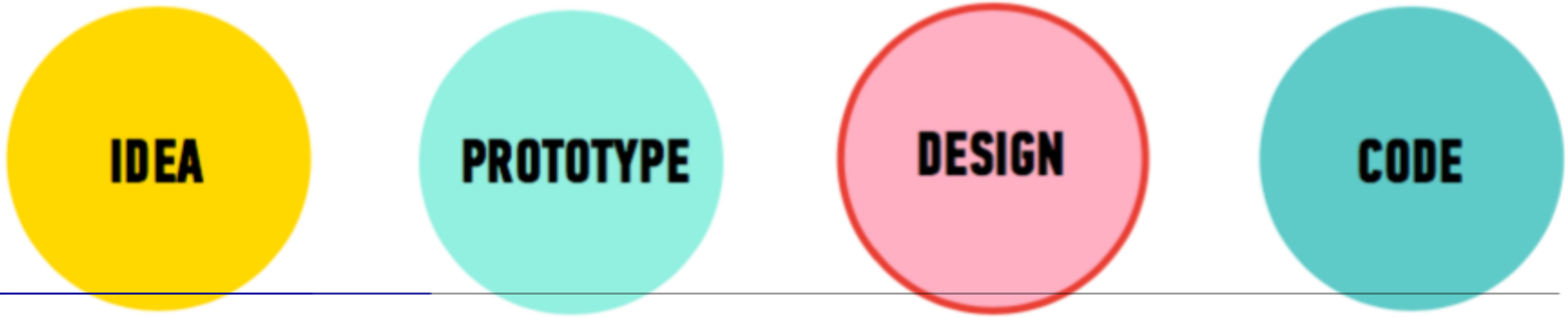
GETTING STARTED



A PICTURE IS WORTH 1000 WORDS:

- How will users most likely access my site?
- What type of content/information is most valuable? How will I structure things in a way that users will be able to easily find the most important information?
- How will users navigate the site? Will it be a single-page or multi-page site?
- What types of organizational structures are possible?

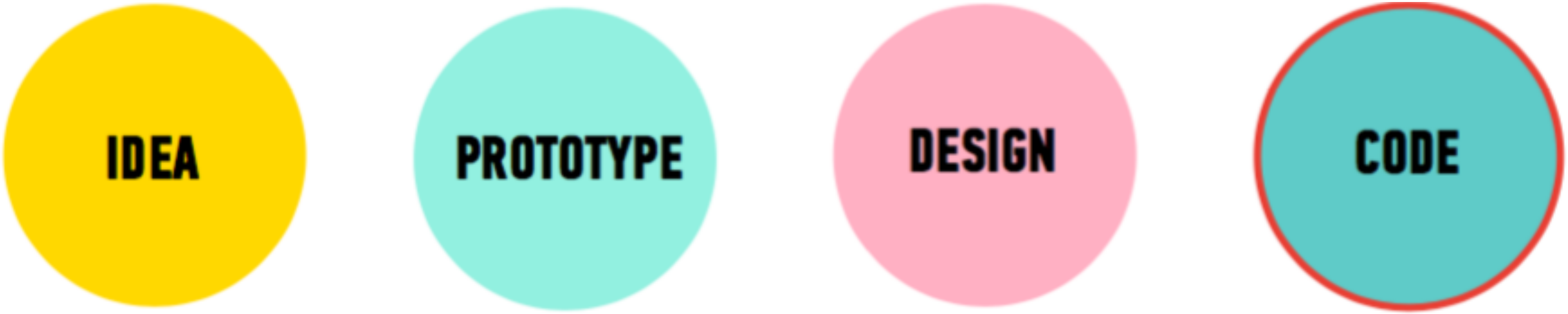
GETTING STARTED



FILL IN THE DETAILS:

- Pick 4-5 colors for the site
- Look through Google Fonts/Font Squirrel and pick out 2-3 fonts for the site
- Use sites like https://unsplash.com/ for high-resolution stock images
- Sites like http://www.siteinspire.com/ , http://www.awwwards.com/ and http://www.webdesign-inspiration.com/ are great for inspiration!

GETTING STARTED



IDEA

PROTOTYPE

DESIGN

CODE

PLAN MORE, CODE LESS:

- Create a features/functionality list
- Start with the structure (HTML), then add styles (CSS), then work on interactions (pseudo code and JavaScript).

TIMELINE

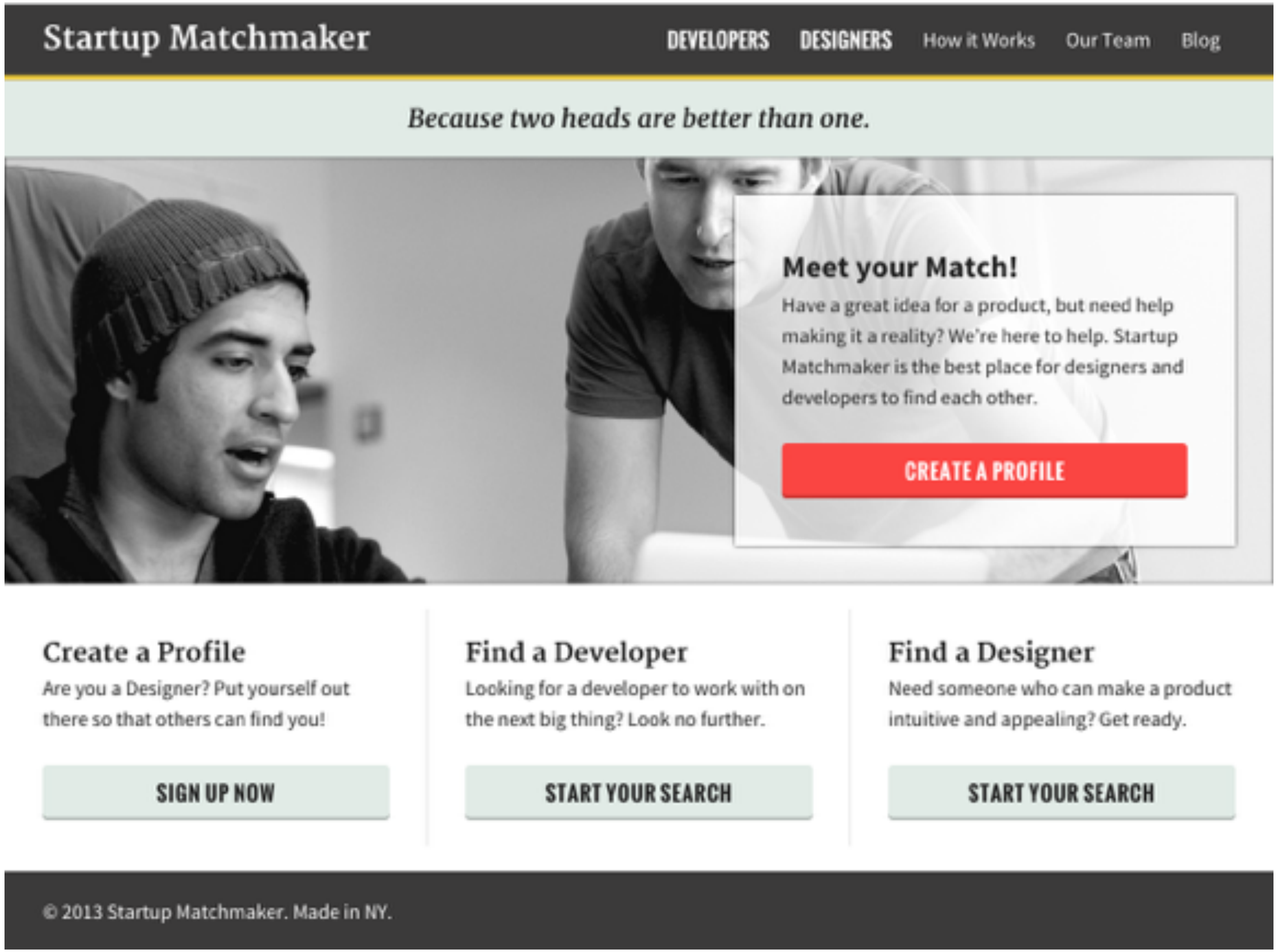
Milestone 1 3/13/16 Project Proposal / Wireframes

Milestone 2 TBD

HOMEWORK

LAB — TRAVEL BLOG

LAB — STARTUP MATCHMAKER



Because two heads are better than one.

Meet your Match!

Have a great idea for a product, but need help making it a reality? We're here to help. Startup Matchmaker is the best place for designers and developers to find each other.

Create a Profile

Are you a Designer? Put yourself out there so that others can find you!

Find a Developer

Looking for a developer to work with on the next big thing? Look no further.

Find a Designer

Need someone who can make a product intuitive and appealing? Get ready.

© 2013 Startup Matchmaker. Made in NY.

HOMEWORK

- **Continue working through the lab we started on today**

PSEUDO CODE

EXIT TICKETS