

Welcome to Bootcamp Prep!

Fullstack Academy

Javascript Review

- <https://bit.ly/FSA-Prework>
- <https://bit.ly/BCP-JS-Review>
 - 6 or higher? Good to go!
 - Lower? Might find course difficult
- **Intro to Coding**
 - <https://www.fullstackacademy.com/intro-to-code>
 - For people w/ no coding experience

Slack + Git

- Our Repo: [Repo Link](#)
- Join Slack: [Slack Group Link](#)

Instructors

- **Sulamita (@sulamita)**
- **Robin (@robin)**



Course Overview: Topics

#	Topic
1	Introduction, Tidy Code
2	Loops, Debugging
3	Coercion and Truthiness
4	Caesar Cypher
5	Scope
6	Arrays I

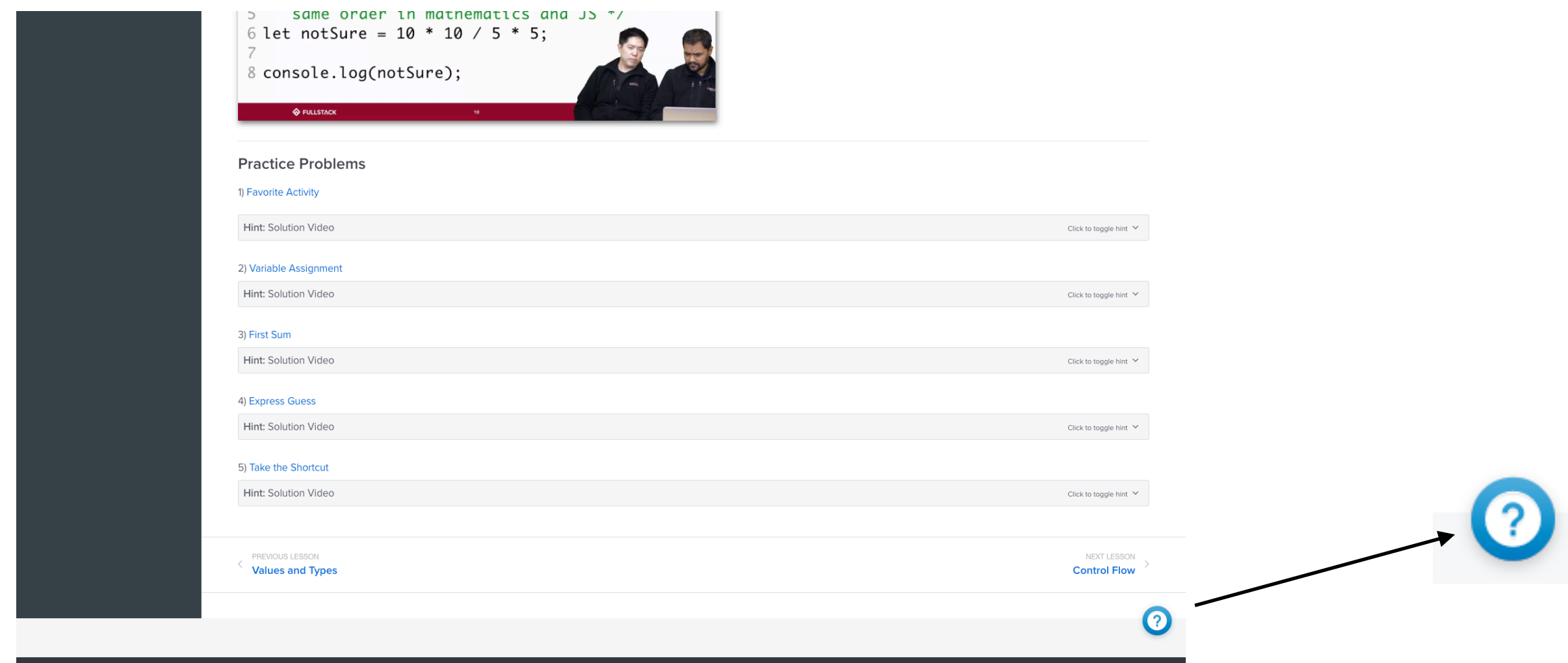
#	Topic
7	Arrays II
8	Sudoku Validator
9	Objects
10	Recursion I
11	Recursion II
12	Wrap up

Course Overview: Format

1. Watch the pre-recorded lecture (if time permits)
2. Attend the live lecture (required)
3. Work on the workshop with your partner (required)
4. Watch the workshop review videos and compare your answers with the provided solutions (strongly recommended)

Workshops

- Workshops are not intended to be finished
- You will be randomly paired with a different partner or small team for each workshop
- Instructors and fellows available via the Help Desk



Workshops: pair programming

- **Pair programming**
 - 2 programmers, sharing the same computer, work collaboratively on all aspects of software development (Williams and Kessler 2000).
 - "Driver" writes code
 - "Navigator" plans code and reviews it as it is written
 - Switch every 20 or 30 minutes
- **At Fullstack, you switch partners for each workshop**

Workshops: pair programming

- **Why use pair programming at Bootcamp Prep? Shouldn't I practice on my own?**
 - "Teach to learn" is a central aspect of Fullstack's instructional philosophy
 - Communicating about code is just as, if not more important, than actually writing it
 - Solving problems with a partner is the best way to prepare for an admissions interview at a top bootcamp
 - Most top bootcamps, and a growing number of employers, use pair programming every day

Course Overview: Projects

- Two projects will be introduced as we progress through the course
- 100% optional
- Good opportunity to use what you're learning to build larger programs

Course Overview: Projects

- Two projects will be introduced as we progress through the course
- 100% optional
- Good opportunity to use what you're learning to build larger programs

Course Overview: Practice Assessments

- We'll take a practice assessment on the last day of class
- It's meant to help you prepare for the real admissions assessment
- A second practice assessment will also be available to you on LearnDot after you finish Bootcamp Prep

Fullstack Community Values

- Be patient. With yourself and others.
- Ask questions (even “dumb” ones)
- Trust the process. Be on time.
- Help others. Teach others.
- We like to have fun, and...

Fullstack Community Values

- Be mindful of:
- Subtle “isms”
- E.g. Sexism, Racism, Ageism
- Be Professional - “No Asshole Policy”
- You are each other’s most valuable resource
- No NSFW content. When in doubt, leave it out.

Why is JavaScript so Popular?

- Powers the majority of web applications
- Runs almost anywhere on the “full stack”: web browsers and servers
- Is relatively learner friendly

ES5? ES6?

- **ES5 and ES6 are different versions of JavaScript**
- **JavaScript versions are backwards compatible**
 - Older features work in newer environments
 - Newer features may not work in older environments
- **Fullstack teaches "ESNext"**
 - Immersive programs continuously updated to cover the latest JS features
 - BCP program covers some newer features too, but only those that are helpful for beginning developers — don't expect in-depth or comprehensive coverage of new JS features in this course

Getting the most from this course

- Complete the BCP Pre-work curriculum first!
- Do the pre-readings and watch the pre-recorded lectures
- Enjoy the "productive struggle"
- Ask for help when you need it (otherwise, why are you here?)
- Don't look at solutions until you give problems an honest try
- Go back to unfinished workshop problems and finish them when time permits
- Watch the solution videos, even for problems you think you got right
- Review the solution code

Tidy Code Tips

Why write tidy code?

- **Easier and faster to debug and read your code**
 - Engineers spend a lot more time reading and debugging code than actually writing it!
- **Demonstrates qualities attractive to bootcamps and employers:**
 - Professionalism: no longer a "beginner"
 - Attention to detail
 - Pride in work



this function is amazing!

Indentation

```
1  // indent one level inside every code block
2  // code blocks start with a { and end with a }
3
4  function amazingFunction() {
5      // inside the function block!
6      if (true) {
7          // inside the if block!
8          console.log('this function is amazing');
9      }
10     // outside of the if block
11 }
12
13 // outside of the function block
14 amazingFunction();
```



Variable Naming

```
1  // use camelCase when defining variable names
2  let myFavoritePlace = 'Fullstack';
3
4  // don't use ambiguous variable names!
5  let x = 68;
6
7  // use names that describe the value they contain
8  let currentTemp = 68;
9
10 // it's ok to use short variable names as counters, like i in a for loop
11 for (let i = 1; i <= 3; i++) {
12   console.log(i);
13 }
14
```