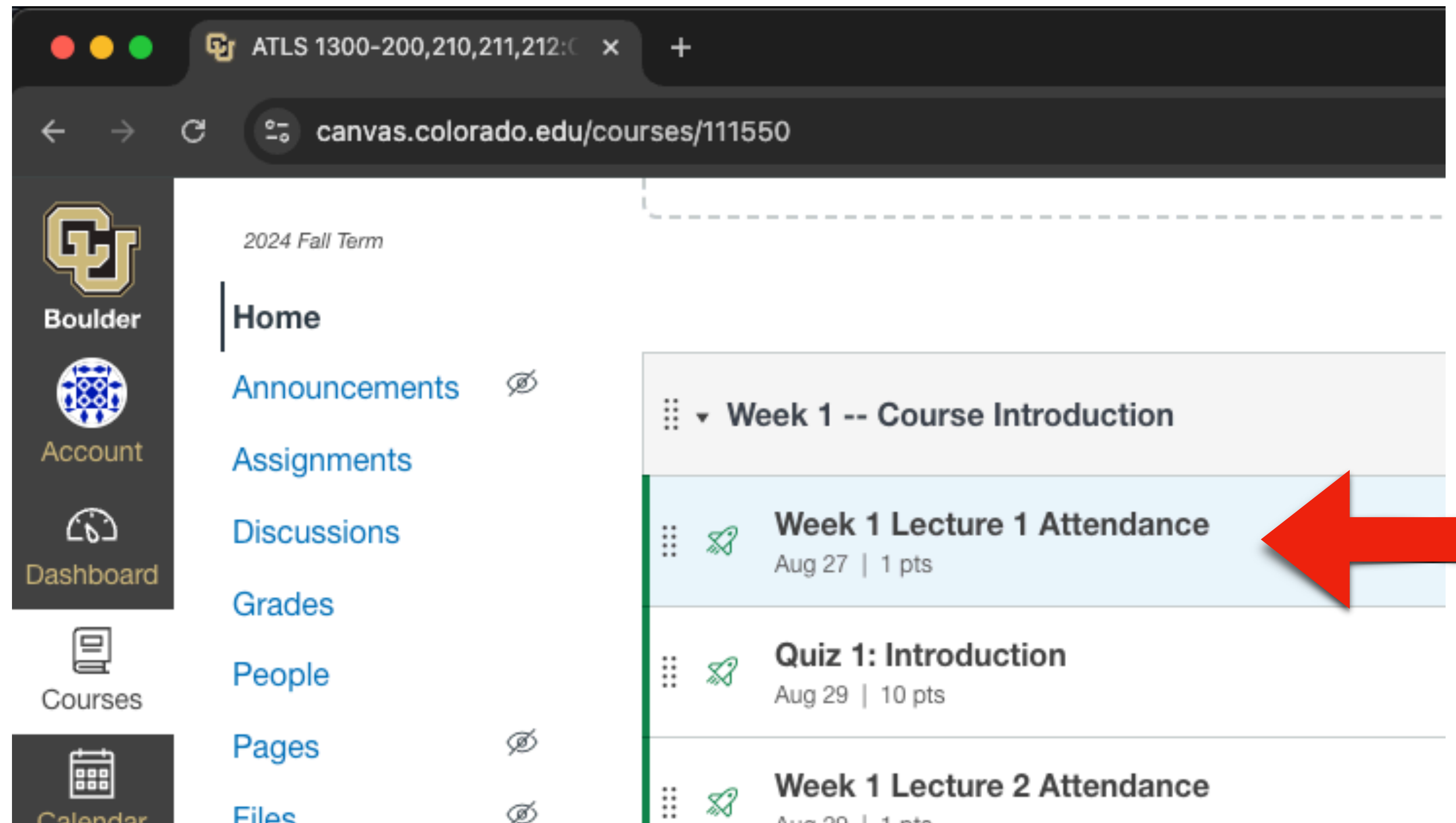


Attendance Is A Thing

I don't like it either

- First thing first - go to Canvas and do the Attendance 'Quiz' that proves that you know the pass code and were also probably here.
- Today's code is:

welcome



CF1

Week 1 - Which Way Is Up?

Gabe Johnson | August 27, 2024

Course Structure

16 Weeks, minus Fall Break

- Two main halves:
 - Throughout weeks 1–8 we will cover the main course content
 - The back half, weeks 9–16, are all about applying those topics to a project
- Weeks have a general cadence:
 - Tuesdays: Guest lecture & “workshop lite” days
 - Thursdays: Content Lecture
 - Fridays: Recitation

	Su	Mo	Tu	We	Th	Fr	Sa	
Aug					1	2	3	
	4	5	6	7	8	9	10	
	11	12	13	14	15	16	17	
	18	19	20	21	22	23	24	
	25	26	27	28	29	30	31	Intro
Sept	1	2	3	4	5	6	7	Animation
	8	9	10	11	12	13	14	Interaction
	15	16	17	18	19	20	21	Loops
	22	23	24	25	26	27	28	Functions
Oct	29	30	1	2	3	4	5	Functions
	6	7	8	9	10	11	12	Objects
	13	14	15	16	17	18	19	Objects
	20	21	22	23	24	25	26	Project & Debugging
Nov	27	28	29	30	31	1	2	Project & Communication
	3	4	5	6	7	8	9	Project
	10	11	12	13	14	15	16	Other Frameworks & Project
	17	18	19	20	21	22	23	Project
	24	25	26	27	28	29	30	Break
Dec	1	2	3	4	5	6	7	Project
	8	9	10	11	12	13	14	Project
	15	16	17	18	19	20	21	
	22	23	24	25	26	27	28	
	29	30	31					

CF1: ATLS 1300

Fall 2024

Gabe Johnson

Tuesdays: Guest Lecture (live or video) + reflection writeup

Thursdays: Core content lecture in class

Fridays: Recitation

Sunday Midnight: Homework (code + reflection writeup)

Notable Days:

Sept 2: Labor Day

Nov 23–Dec 1: Fall Break

Nov 7: Project A Presentations

Dec 12: Project B Presentations

Front Half & Back Half

Front Half

Main Course Content

- We'll learn p5.js over eight weeks
- This is enough to make interactive graphics programs
- Each week = a different topic
- They build on one another

Back Half



Project Work

- Apply the topics we learned in the front half
- ... by working two projects
- Each will have a *proposal* that is peer reviewed
- Project A is solo, Project B is with a group
- Project B can be a continuation of someone's Project A, if your group likes
- Each project culminates in a presentation
- Your communication of each project is as important as the code!

Weekly Cadence




Tuesdays

Guest Lecture & Workshop Time

- Half the time, Tuesday we will have a guest lecturer in class
- Other half of the time, we'll have a video link from a guest lecture from the other section of CF1
-  Attendance is taken Tuesdays when there is a guest lecture in our section
- Every guest lecture has an associated  *Reflection Assignment (due Thursday)*
- Tuesday always closes out with workshop time
- Class is over early if nobody is left


Thursdays

Content Lecture

- I'll give a talk that covers the same content as the CodingTrain lectures
- You're expected to watch videos *before* class
 - The  Content Quiz (in Canvas) is due *before* class
-  Reflection on the Tuesday Guest Lecture is also due before class on Thursday.
- Content links will be available on Canvas and on my CF1 Github Repo
 - <https://github.com/johnsogg/cf1>
 - You should bookmark this repo as you'll need to reference it often
-  Attendance is taken; this is part of your grade

Fridays

Recitation

-  Attendance is taken here too and also part of your course grade
- This is a good chance to get help from your LA and fellow students
- Coding can be social! You'd be amazed at how much you learn by sitting with others and watching how they work. Give and get feedback!
- The LAs leading our section's recitations are: Gia, Wolfgang, and Lindsey
 - If your TA isn't mentioned there, let me know

Grades

Grading Scheme

Note: no tests!

Attendance & Participation	10%
Quizzes	10%
Guest Lecture Reflections	25%
Weekly Homework & Reflections	25%
Project A	20%
Project B	20%

Grade Ranges

94	100	A
90	94	A-
87	90	B+
84	87	B
80	84	B-
77	80	C+
74	77	C
70	74	C-
60	70	D
0	60	F

How To Win At CF1

Course Goals

This is about getting *intuition* around how software is made

- Baseline, this is about learning to code
- But this is really about learning how to learn to code
- ... and how to think about structuring problems so you can write software to address them
- This course is meant to help you answer the question “is code my thing?”
- No shade if it isn't! This is a 4 month course, not a lifetime commitment.

Learning

- The software world moves fast. You have to be able to learn new concepts quickly.
- This is not an after-school special. Characterize what you don't know, and once you've figured it out, characterize the strategy that worked. Remember, re-use, iterate, improve your learning strategies.

Code, Design, and Communication

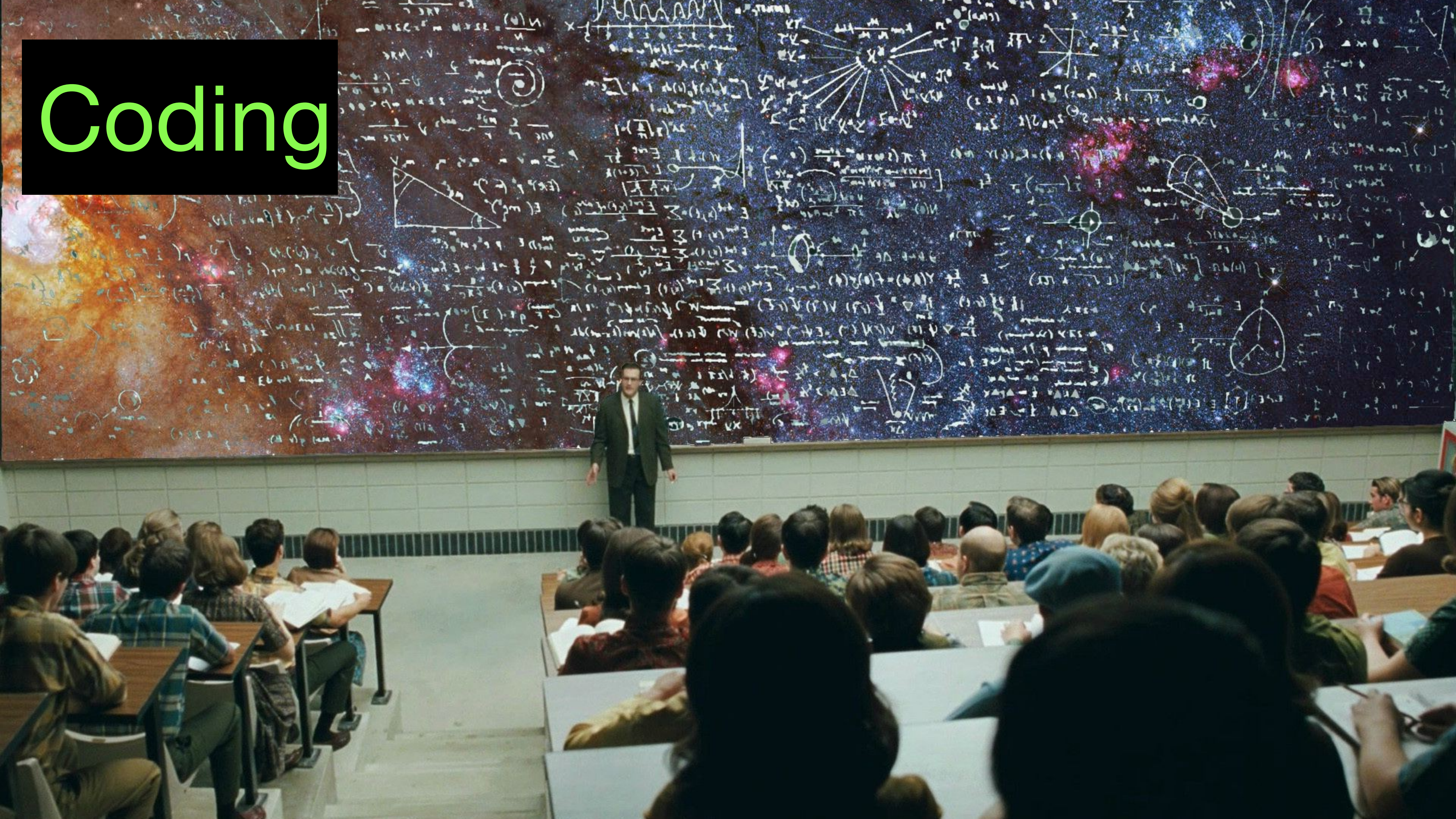
- This is about *design* as much as it is about writing code
- Design is as much about *understanding and framing problems* as it is *solving* problems.
- And if you hope to inform, sway, persuade an audience, you need to know how to communicate your work.
 - For your projects: Have a story about **why** you built what you built. We can see what it does but we might not understand the motivations and background.
 - Your project grade will place a heavy weight on how you present it.

Collaboration

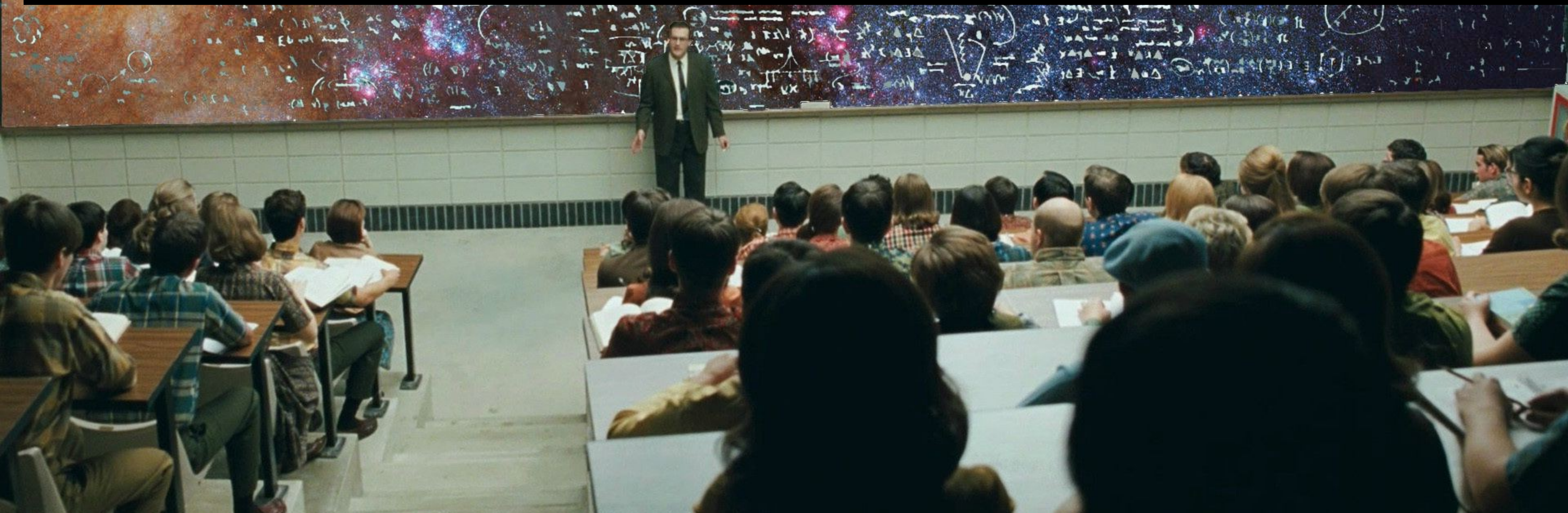
Do! Strongly Encouraged!

- Work with your peers, friends, family, bitter enemies!
- Document all Internet help and student interaction
- **Caveat:** *type everything yourself*
 - This is important. Especially at this phase of your software career.
 - This means: never copy/paste random code from the internet. It won't work and you won't know why.

Coding



Computer science is full of people who love to wield their arcane ability over other people, using it as a weapon to make others feel inept. Those people make our job as learners and educators more difficult because they reinforce the wrong idea that programming is hard, and that they are smarter than you. They make the learning environment hostile.



Coding

- This isn't a computer science class necessarily, but we'll be touching on some of those topics.
- Writing code can be frustrating!
- I assure you it is awesome, once you get the hang of it.
- Perseverance! You will eventually win!
- If you're lucky enough to "get it" easily, **don't be macho about it.** Don't wield your skills as a weapon. Try to help others see things the way you do.

Self Indulgent About Your Instructor Section

Gabe Johnson



- Undergrad in Computer Science (from here at CU!) ~2002
- PhD in Computational Design from Carnegie Mellon ~2012. My thesis advisor was Mark Gross, who is the director of Atlas now.
- Startup to commercialize my thesis on sketch recognition. We went through Techstars in 2017 and have a video to prove it.
- Taught in the CS dept: CSCI 1300 and 2270, and more recently the postbacc version of 2270 (Data Structures & Algorithms)
- Professionally a software engineer, mostly focused on UX and frontend because I like to make pixels do interesting things for real humans.

Gabe Johnson



- You can call me Gabe. I'm not really into rank.
- Professor, Doctor, Hey You - these all work as well if you're more comfortable with that.
- That's what I looked like maybe 10 years ago? Nobody ever takes pictures of me so that's the best I got.

Homework & Reflections

Homework

Use the p5.js web editor

- The p5.js web editor is at <https://editor.p5js.org/>
- I recommend creating a Github account at <https://github.com/>
 - Github is probably the most popular code sharing system
 - It adds lots of great UI & tooling to the `git` source code management tool
- Then use your Github account to sign in to the p5.js editor so you can save and share your code sketches.
- You will probably use Github in this course, and if you do any sort of software development in the future you'll almost certainly need a Github account for that.

Reflection Documents

This goes for homework & guest lecture reflections, etc.

- For each homework and guest lecture we'll ask you to upload a *reflection* on what you just experienced (or endured, depending on your POV)
- Write thoughtfully, in complete sentences. Proofread. Be professional. You can be funny and let your own rad personality shine through! But it must be coherent and communicative.
- Upload **PLAIN TEXT** only. Markdown is OK because it reads like plain text.
 - No PDFs
 - No Google Docs
 - No Links
 - For the love of all that is pure and good, No MS Word

Due Dates

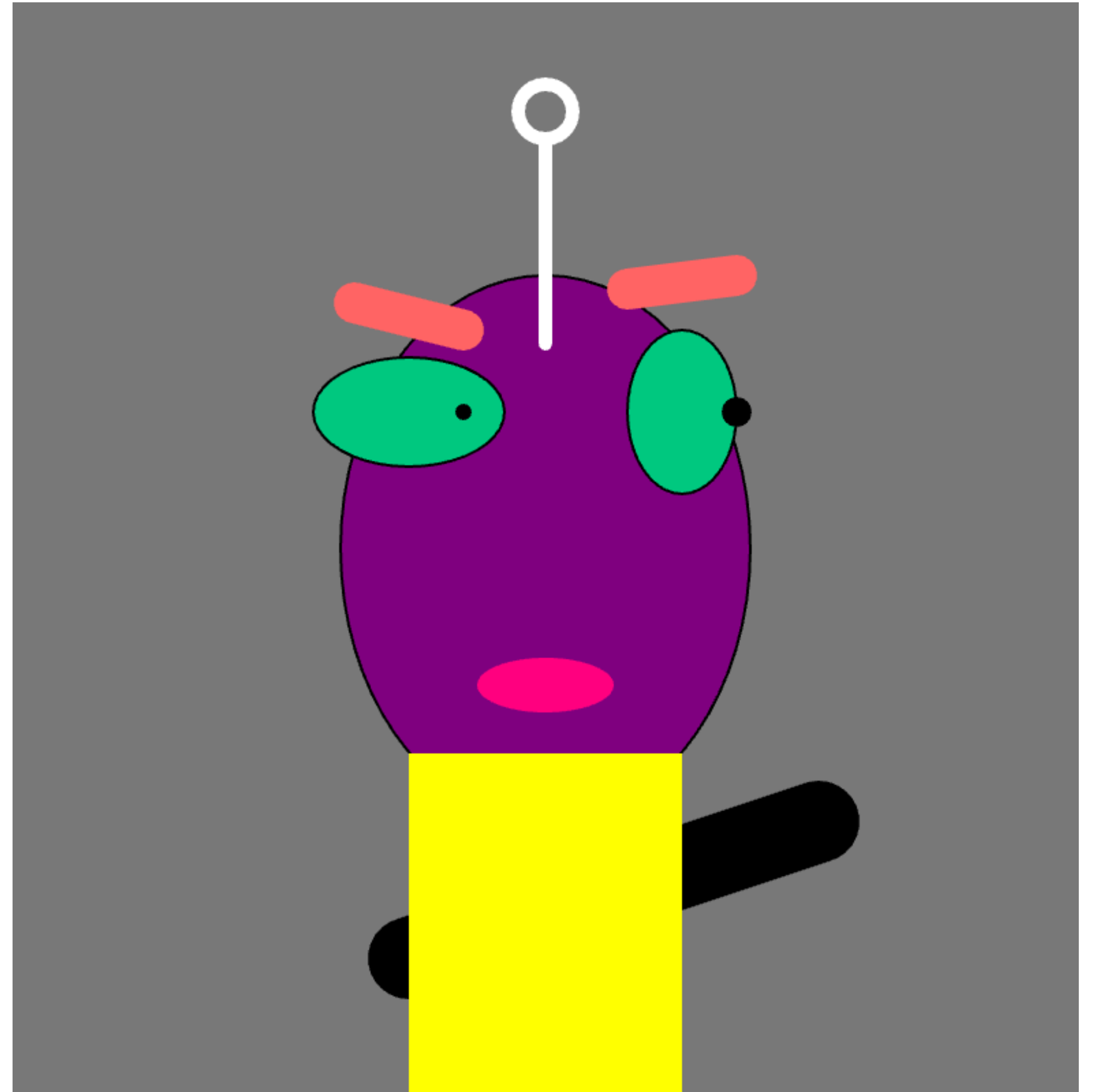
- Homework code and reflections will be due on Sunday night, 11:59pm
- Content quizzes are due just before the Thursday content lectures, 2:00pm
- Guest lecture reflections will also be due just before content lectures, 2:00pm
- Attendance quizzes are only open during the first 15 minutes of class

Homework 1

Homework 1

Code a Self-Portrait in p5.js

- Watch the Coding Train 1.x videos
- And maybe also read the O'Reilly book
- Details are in Canvas
- Have fun!



```
1 function setup() {  
2   // Create drawing canvas  
3   createCanvas(400, 400);  
4  
5   // Dark gray background  
6   background(120);  
7  
8   // Purple face  
9   fill(127, 0, 127);  
10  ellipse(200, 200, 150, 200);  
11  
12  // Green eyes  
13  fill(0, 200, 127);  
14  ellipse(150, 150, 70, 40);  
15  ellipse(250, 150, 40, 60);  
16  
17  // Black pupils  
18  fill(0);  
19  ellipse(170, 150, 5, 5);  
20  ellipse(270, 150, 10, 10);  
21  
22  // Orange eyebrows  
23  stroke(255, 100, 100);  
24  strokeWeight(15);  
25  line(130, 110, 170, 120);  
26  line(230, 105, 270, 100);  
27  
28  // Antenna  
29  stroke(255);  
30  strokeWeight(5);  
31  line(200, 125, 200, 50);  
32  // No fill  
33  noFill();  
34  ellipse(200, 40, 20, 20);  
35
```

