

# Welcome to CS571!

# Building User Interfaces

**Cole Nelson**

# What is **CS571** about?

→ UX Development

...but what is UX Development?

*The realm of the UX developer exists somewhere between that of the traditional developer and the designer. We're not really designers, yet to be a good UX developer you certainly need to have an eye for design. In the same vein, we're not traditional developers but we certainly need to have development experience and expertise. Often this experience spans multiple technologies, languages, and platforms.*

— Tim R. Todish

*It falls on the UX developer to bridge the gap between design and technology. We need to be able to think and speak the language of designers. It's our job to help translate their vision to the development team in a way that they can understand and accept. This can be a critical piece of the puzzle in a project, especially if the design and the interactions behind it are complex.*

— Tim R. Todish

*Similarly, we need to speak on behalf **the developers** to help reign in the designers, at times. If they are coming up with concepts that will be extremely difficult or time consuming to implement, we can explain the limitations of the technology and the complexity involved in implementing their designs, and try to come up with an acceptable alternative.*

— Tim R. Todish

# UX Development

= Software Engineering + UX Design

# What does a Software Engineer do?<sup>1</sup>

<sup>1</sup> Kantega | The Mysterious Life of Developers



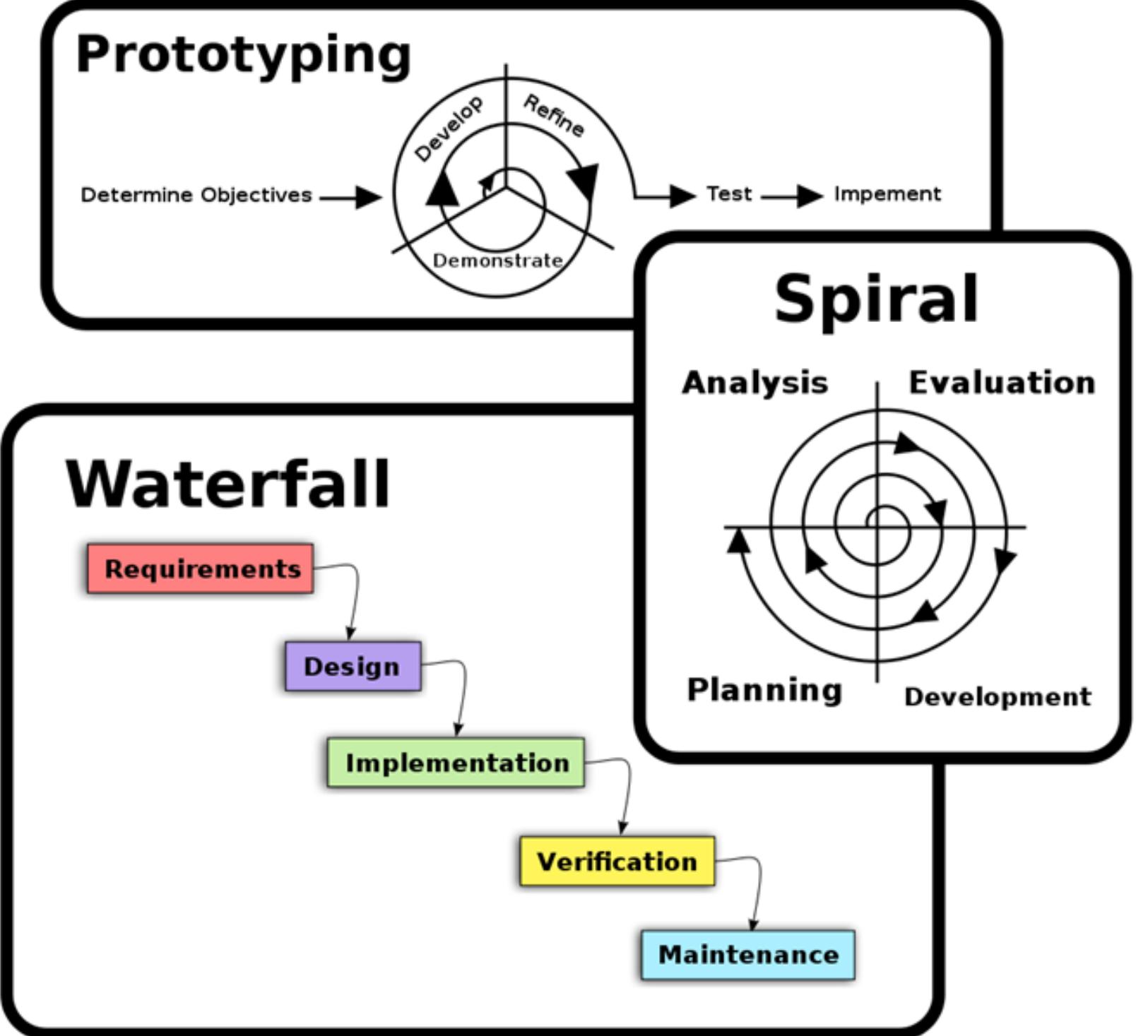
which is devoured in a  
large quantity.

**Definition:** A software engineer is a person who applies the principles of software engineering to the design, development, maintenance, testing, and evaluation of computer software.

But how does a Software Engineer go about this?<sup>2</sup>

<sup>2</sup> What is a Software Engineer?

# Software Development Lifecycle<sup>3</sup>



<sup>3</sup> SDLC Process

# What does a UX Designer do?<sup>4</sup>

<sup>4</sup> Kantega | The Mysterious Life Of UX Designers



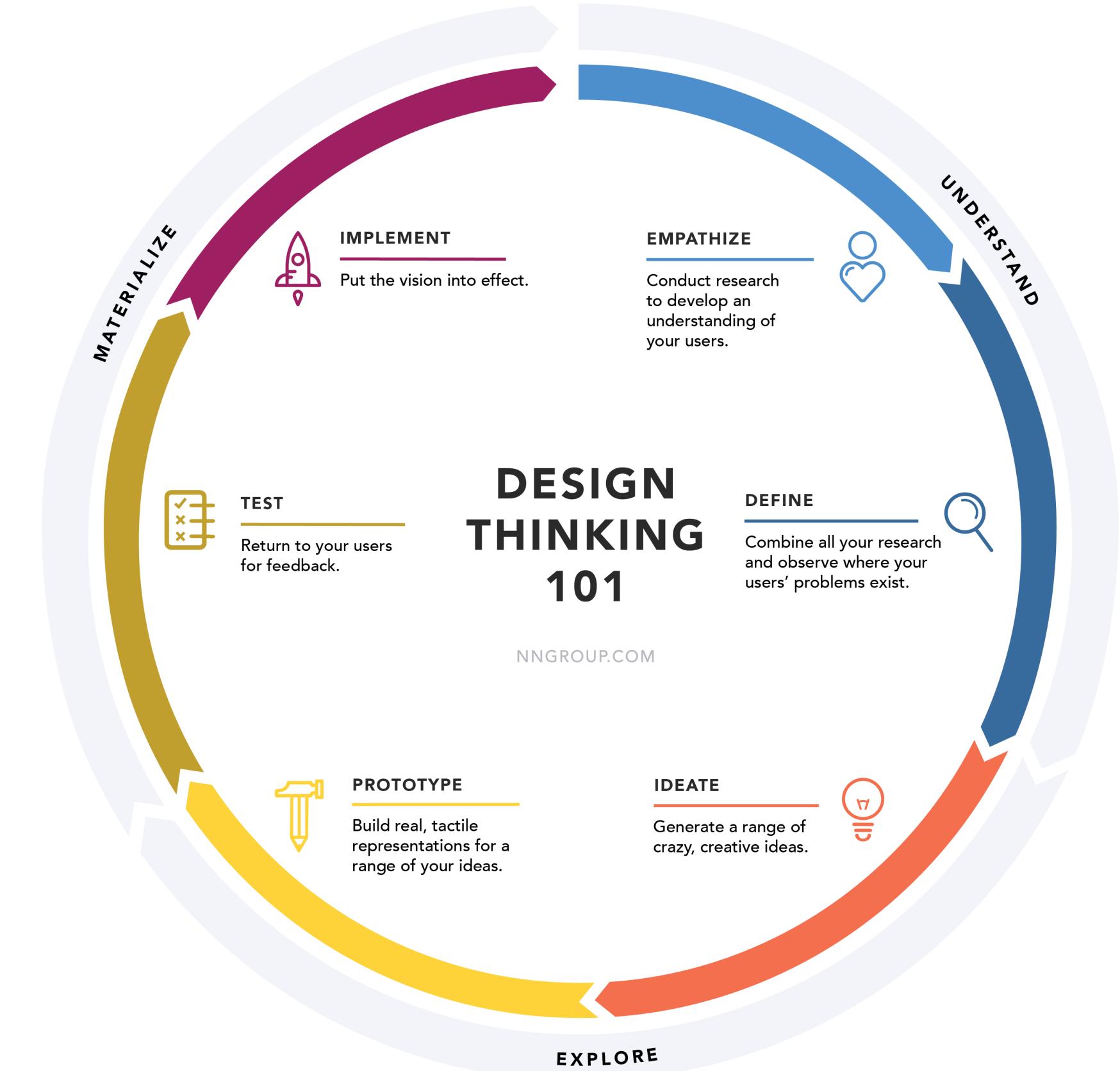
**Definition:** User experience (UX) design is the process that design teams use to create products that provide meaningful and relevant experiences to users. A UX designer is concerned with the entire *process* of acquiring and integrating a product, including aspects of branding, design, usability, and function.<sup>5</sup>

How does a UX designer go about this?

<sup>5</sup> Interaction Design Foundation

# UX Design Process<sup>6</sup>

Looks familiar?



<sup>6</sup> NN/g Design Thinking

# So then, what is UX Development?<sup>7</sup>

<sup>7</sup> Don Norman: The term "UX"

A portrait of an elderly man with a white beard and glasses, wearing a blue denim shirt. He is pointing his right index finger towards the camera with a slight smile.

NN/g

**Don Norman:**  
**The Term UX**

# UX Development Trajectories

- One-person development team to build full-stack applications
- A developer who speaks the language of the designers
- A designer who can also build native prototypes
- A bridge/translator between designers and developers in large/complex organizations

# Course Details

# Instructor



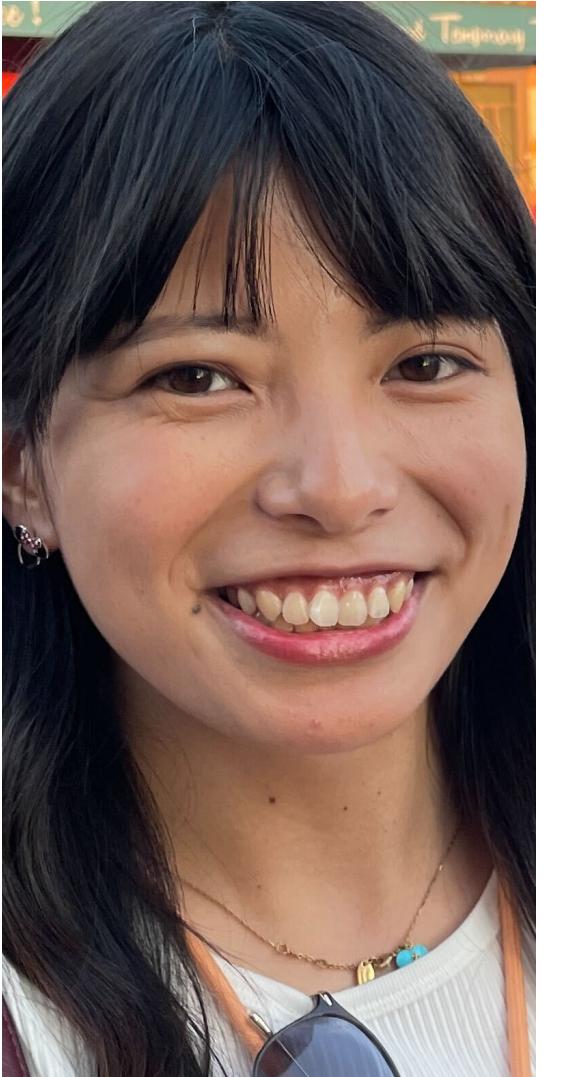
Cole Nelson  
*ctnelson2@wisc.edu*

B.S. Software Engineering  
M.S. Computer Science

<https://github.com/ctnelson1997>  
<https://cs571.org/>

# TAs

Amy Koike



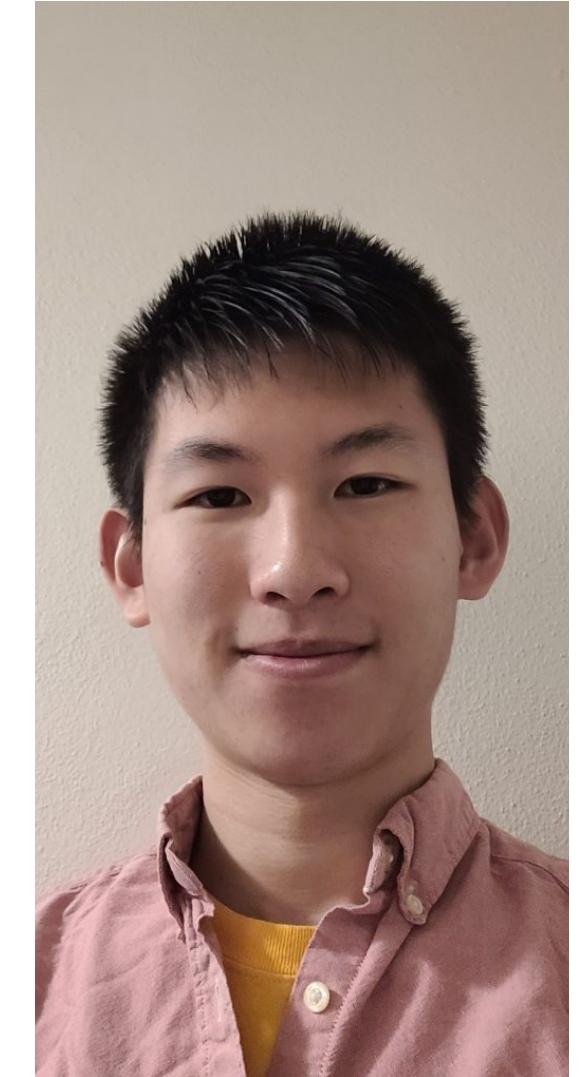
Hongtao Hao



Suhas Kurukuri



Leo Cui



# Peer Mentors

- Wai Linn
- Josh Charpentier
- Catherine Tan
- Rainy Jin
- Kruthiventi Nikhil
- Zhaoyang Liu
- Jerry Yu

# Who are you?

We'll get to know you in HWo! This information will be shared with your classmates in HW2 & HW4, BadgerBook.

## Alan Turing

### Computer Science

Alan is taking 12 credits and is from Wisconsin.

They have 4 interests including...

- Reading
- Lifting
- Running
- Sudoku

# Course Information

# Course Basics

- Meet Tuesdays & Thursdays; lectures will be recorded and posted to Kaltura.
  - In-person attendance is *encouraged*.
- Tuesday's lecture will cover programming topics, Thursday's lecture will cover interaction design topics.
- Bring a laptop!

<b>Week Of</b>	<b>Topics</b>	<b>Week Of</b>	<b>Topics</b>
Jan 22	Intro to CS571 & Design Thinking	Mar 18	Mobile Dev 1 & Prototyping
Jan 29	Web Dev Basics 1 & Web Dev Basics 2	Mar 25	<b>Spring Break!</b>
Feb 5	Web Dev Basics 3 & Visual Design	Apr 1	Mobile Dev 2 & Mobile Design
Feb 12	Web Dev 1 & Web Design	Apr 8	Mobile Dev 3 & VUI Design/Exp Prototyping
Feb 19	Web Dev 2 & Interaction Design	Apr 15	Speech Dev 1 & Designing Agents
Feb 26	Web Dev 3 & Expert Evaluation	Apr 22	Speech Dev 2 & User Evaluation
Mar 4	Web Dev 4 & Accessibility	Apr 29	Fullstack Development & Professor's Choice
Mar 11	Web Dev 5 & <b>Midterm Exam</b>	May 10	<b>Final Exam</b>

# Point Distribution

Item	Points	Notes
Weekly Assignments	45	10 HWs (4 pts) & 3 HWs (1-2 pts each)
Weekly ICAs	10	12 ICAs, 1 pt each, 2 lowest dropped
Midterm Exam	20	Double-sided notesheet
Final Exam	25	Double-sided notesheet

# Grading Scale

**A                  AB                  B                  BC                  C                  D                  F**

---

[100, 94]    (94, 88]    (88, 82]    (82, 76]    (76, 70]    (70, 60]    (60, 0]

# **Weekly Assignments & ICAs**

- Keep up with the lectures & course content!
  - Concepts build throughout the semester.
- Every weekly assignment is released on Tuesday and due the following Monday at 11:59 pm.
- ICAs are started in-class on Thursdays and due within 24 hours.

# Midterm & Final Exam

- Exams will be held *in-person*.
  - **You must be on campus in order to take the exams!**
- Midterm Exam is on Thursday, March 14th @ 5:45 pm
- Final Exam is on Friday, May 10th @ 5:05 pm
- See Canvas for past exams.

# Late Policies

- Weekly ICAs may *not* be submitted late.
- Weekly assignments may be turned in *up to* one week (7 days) late with the following restrictions:
  - Each late day is 10% off the assignment grade.
    - E.g. a score of 4.5/5.0 two days late is a 3.6.
    - Days are *rounded up*, e.g. 12:01 am is a late day.
  - The first 10 late days used are penalty-free.
    - These are intended to be used in case of sickness or extenuating

# Academic Integrity

- **Do not share code with others!**
- **Do not use code from previous semesters!**
- You may *discuss* assignments with others, but all work must be done individually.
- Snippets taken from StackOverflow, ChatGPT, etc. **must be cited** with a comment.
- We use tools like MOSS... Don't risk it!

# Course Tools

- Assignments will be submitted via *GitHub Classroom*.
  - Attend GitHub workshop if you are new to Git.
- Questions will be asked via *Piazza*.
- Personal matters will be handled via *email*.

# Workshops

These are extra sessions led by your TAs! They are optional but enhance your learning.

- GitHub Workshop (Week 1) - Leo
- Figma Workshop (Week 2 or 3) - Amy
- Deployment Workshop (Week 8 or 9) - Hongtao
- HCI Workshop (Week 13 or 14) - Suhas

# Office Hours<sup>8</sup>

**Don't just show up, use OHWL to queue.**  
Made by a UW student using React!

Join w/ code 0V7HJHV0

**OH Start Next Week**

**CS571**

Building User Interfaces - Fall 2023

[Back](#)

 Student Code: Q3C83PIE

**Office Hours**

**Monday Office Hours**

 0

Start: Sun, 9/10/2023, 2:00 PM CDT

End: Sun, 9/10/2023, 4:00 PM CDT

Repeats Weekly

Location: [Meeting Link](#)

<sup>8</sup>[Office Hours Schedule](#)

# FAQs

All class-related questions, including grading, deadlines, etc.

→ E-email `cs571@cs.wisc.edu`

Questions about personal matters.

→ Contact `ctnelson2@wisc.edu`

# Badger IDs

The screenshot shows an email interface with the following details:

- Subject:** CS571 New Badger ID
- From:** uw.cs571@gmail.com
- To:** Cole Tyler Nelson (ctnelson2@wisc.edu)
- Date:** Sun 9/3/2023 6:13 PM
- Message Content:**

Hello ctnelson2@wisc.edu,

A new Badger ID has been generated for you:

bid\_1acbb5 [REDACTED]

In order to use the CS571 APIs, you will need to enter this Badger ID at [cs571.org](https://cs571.org) > BadgerAuth > Use BadgerID.

Sincerely,  
The CS571 Course Staff

CS571 course staff and information systems will NEVER send you links or ask for personal information.
- Buttons:** Reply, Forward

This is a valid email! :) We will cover it in JS lectures.

# Software & Downloads

- **Visual Studio Code** for JS Development
- **Node 20 and NPM 10** for React Development
- **Postman** for API Exploration
- **Git** for Version Control/Submission

# McBurney & Accommodations

The University of Wisconsin-Madison supports the right of all enrolled students to a full and equal educational opportunity.

Please inform me of your need for instructional accommodations during the beginning of the semester, or as soon as possible after being approved for accommodations.

# Mental Health

UHS offers no-cost services in psychiatry, individual, couple, and group counseling, alcohol and drug abuse, and other mental health needs.

Schedule an access appointment at 608-265-5600 option 2 or contact 24/7 crisis support at option 9.

You matter!

# Today's Topic

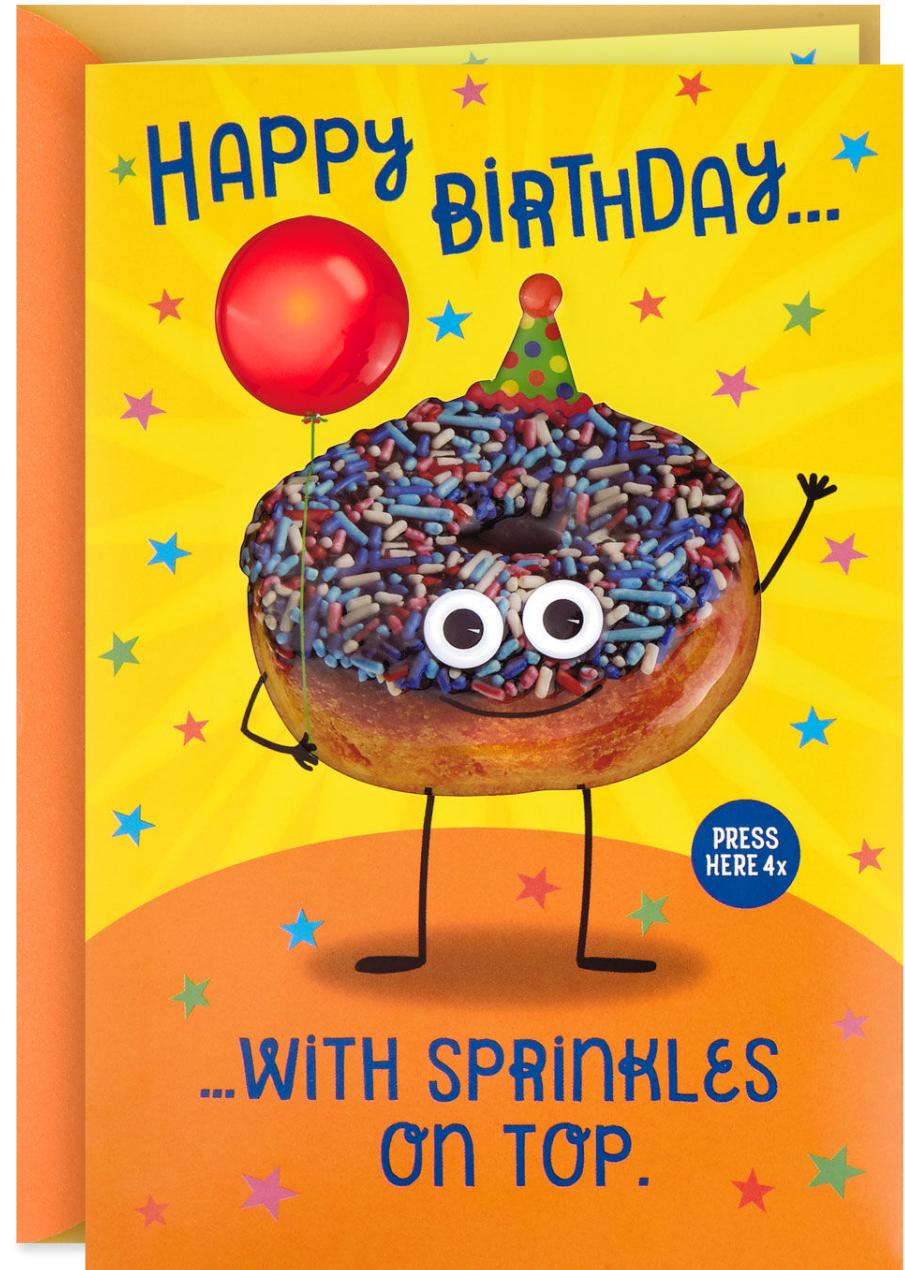
The Web.

# A Website<sup>9</sup>

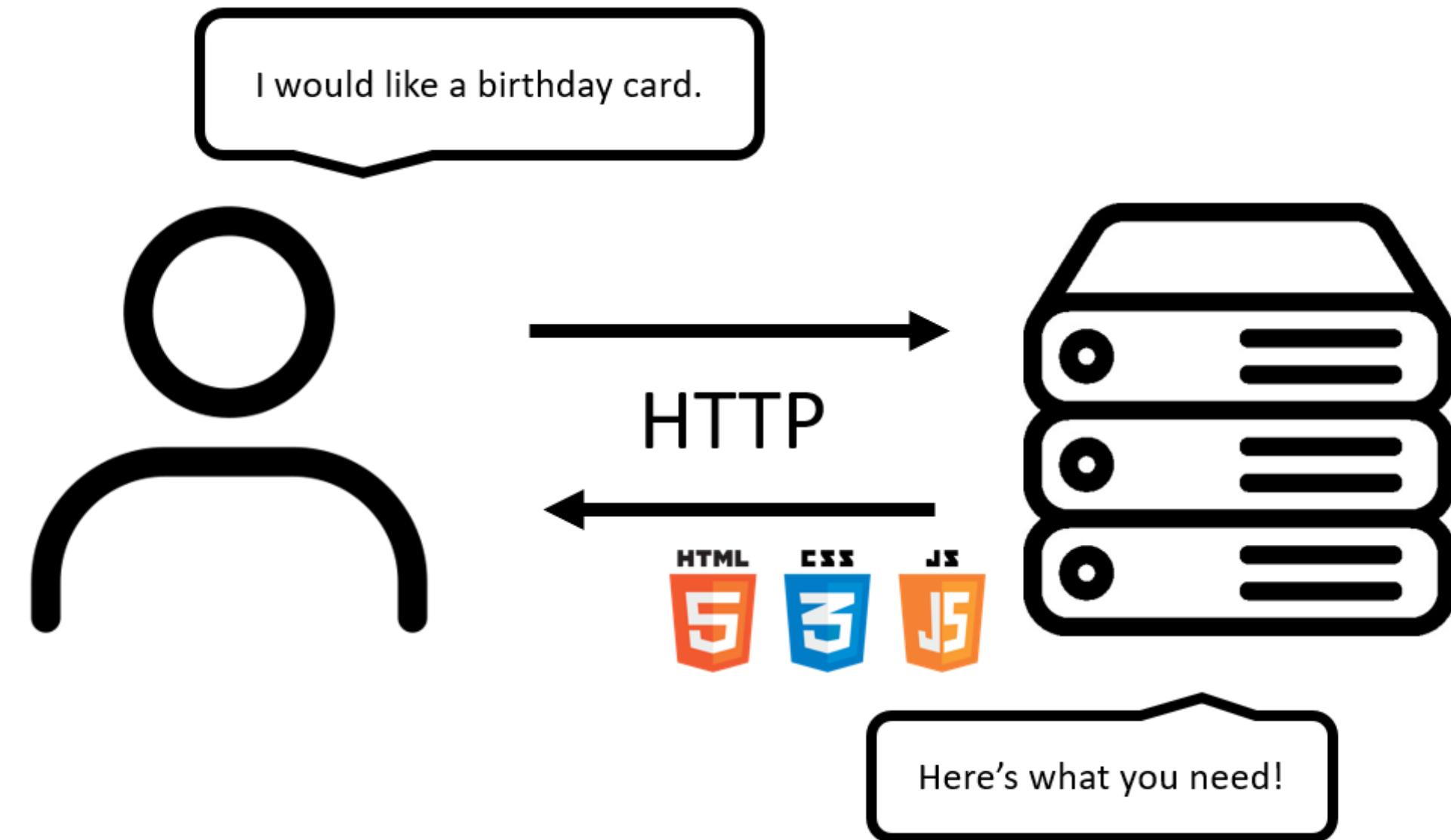
It's like a birthday card!

Concept	Behavior
HTML	Structure
CSS	Design
JS	Behavior

How does it get delivered? HTTP!



<sup>9</sup> Personal Website [Codepen](#)



<sup>10</sup> Server Icon by Pixel Perfect

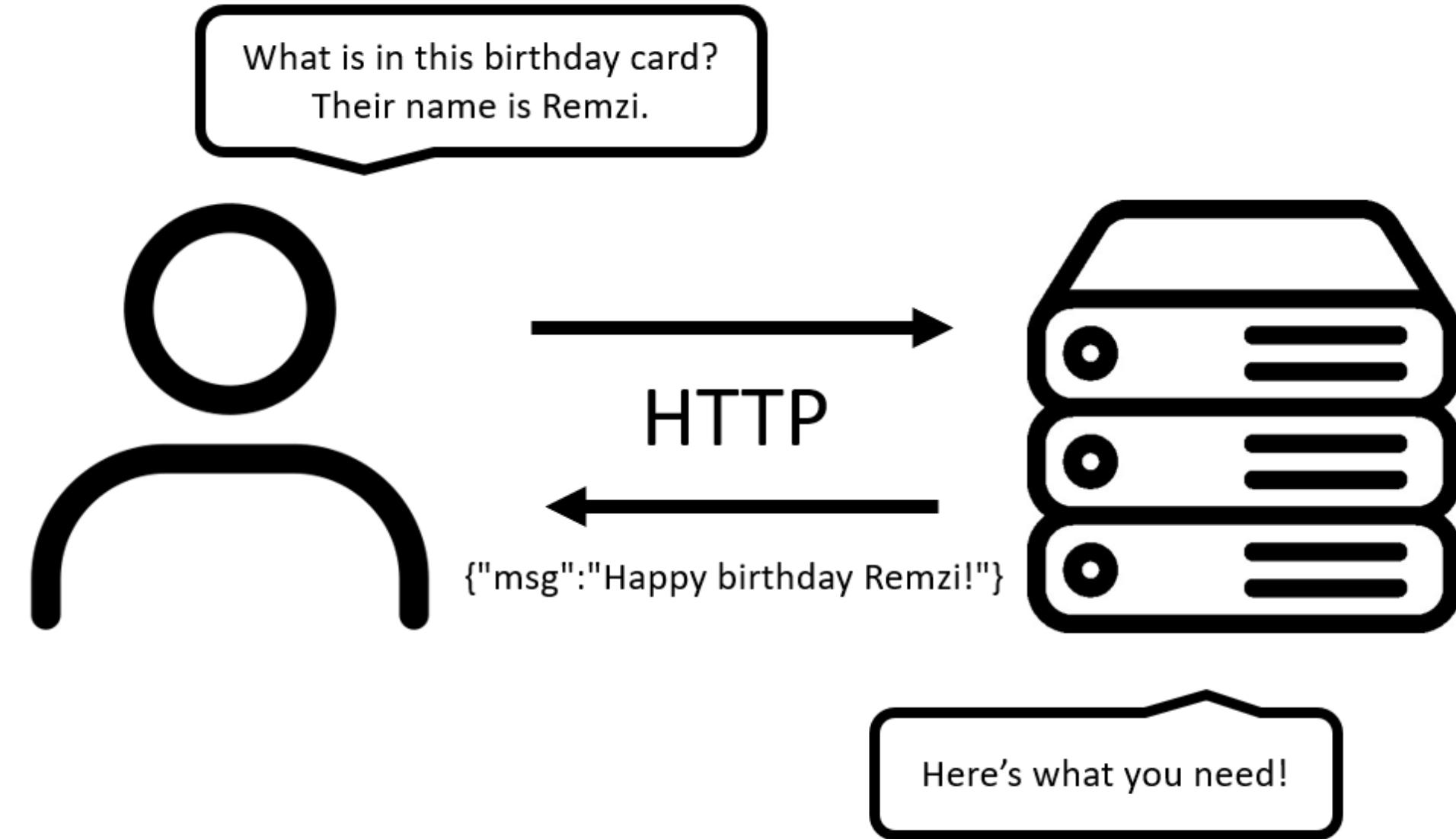


There's nothing inside of the card?!

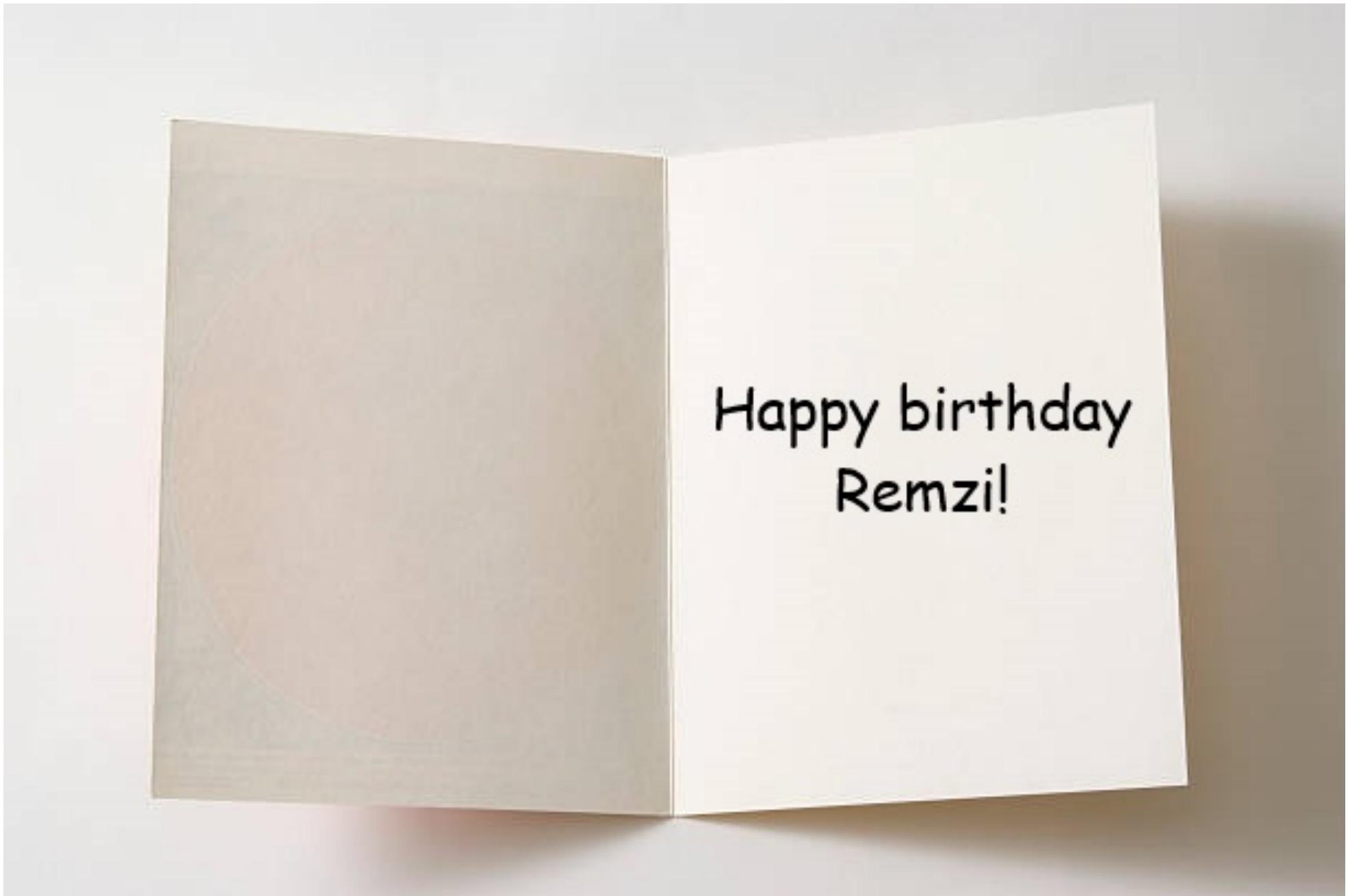
# Communication

We need a *structured way* to communicate data. We'll use **JSON**.

```
[  
 {  
   "id": 1,  
   "title": "New interactive mural invites  
             exploration and engagement with  
             science",  
   "img": "https://www.coletnelson.us  
          /cs571/f22/hw7/api/news/images  
          /exhibit",  
   "tags": [  
     "science",  
     "entertainment"  
   ]  
 },  
 {  
   "id": 2,  
   "title": "Sewer sleuthing, air
```



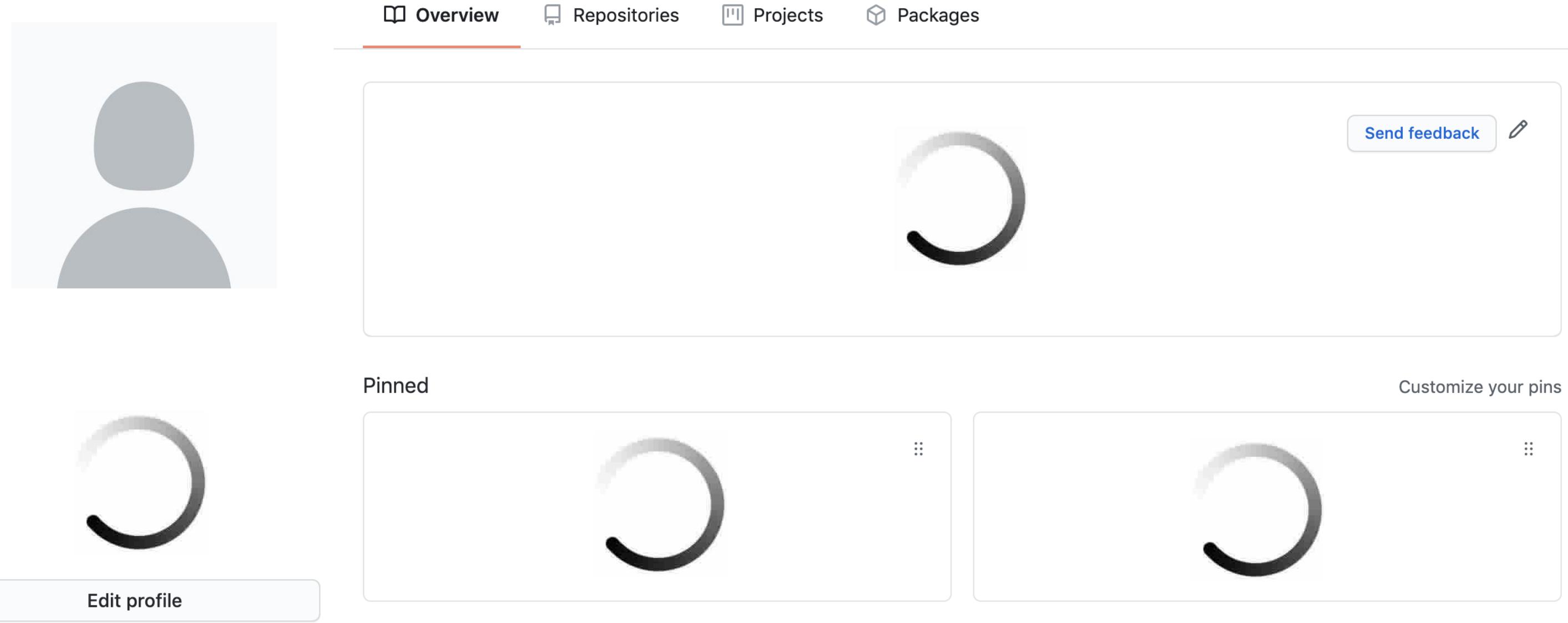
<sup>10</sup> Server Icon by Pixel Perfect



Better!

# Real-World Example

How does GitHub display profile webpages?



What can be displayed with HTML/CSS/JS on load...

```
HTTP/1.1 200 OK
Server: GitHub.com
Date: Fri, 20 Jan 2023 21:36:04 GMT
Content-Type: application/json; charset=utf-8
Cache-Control: public, max-age=60, s-maxage=60
Vary: Accept, Accept-Encoding, Accept, X-Requested-With
Last-Modified: Fri, 20 Jan 2023 12:09:09 GMT
{
  "login": "ctnelson1997",
  "id": 37457808,
  "node_id": "MDQ6VXNlcjM3NDU3ODA4",
  "avatar_url": "https://avatars.githubusercontent.com/u/37457808?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/ctnelson1997",
  "public_repos": 6,
  "public_gists": 1,
  "followers": 3,
  "following": 3,
  "created_at": "2018-03-16T20:56:14Z",
  "updated_at": "2023-01-20T12:09:09Z"
}
```



Response Code

Response Headers

Response Body

JSON data received!



## Mona Lisa Octocat

octocato

Hi, I'm Mona 🙌 You might recognize me as @github's mascot 🦸‍♂️ 😄

[Edit profile](#)

Overview    Repositories    Projects    Packages

octocato / README.md

Hi there 🙌

- 🔭 I'm currently working on something cool!
- 🌱 I'm currently learning with help from [docs.github.com](#)
- 💬 Ask me about GitHub

[Send feedback](#)

Pinned

Customize your pins



atom

Forked from atom/atom



The hackable text editor



JavaScript



vscode

Forked from microsoft/vscode



Visual Studio Code



TypeScript

# Why JSON?<sup>11</sup>

- easy to understand
- human-readable
- language agnostic
- easily convertible to JS objects

```
[  
 {  
   "id": 1,  
   "title": "New interactive mural invites  
             exploration and engagement with  
             science",  
   "img": "https://www.coletnelson.us  
          /cs571/f22/hw7/api/news/images  
          /exhibit",  
   "tags": [  
     "science",  
     "entertainment"  
   ],  
   {  
     "id": 2,  
     "title": "Sewer sleuthing, air
```

<sup>11</sup>Try json

# JSON Basics

- Valid value types include...
  - string e.g. "hello world"
  - number e.g. 24
  - boolean e.g. false
  - array e.g. [1, true, 2, 4.2, "smith"]
  - object e.g. {"name": "Carl", "age": 24}
  - null
- Values can be nested!

# How about design?

**Concept**

---

HTML

---

CSS

---

JS

**Behavior**

Structure

---

Design

---

Behavior

# An Example<sup>12</sup>

Consider this profile page/component

What do you think comprises *design*?

What about *behavior*?

We'll see in later design lectures that appearance, type, behavior, etc. are all part of *design*.

→ Let's see how these are expressed in code.

<sup>12</sup> [CodePen](#)

The image shows a profile page with a yellow header. In the top right corner is a circular profile picture of a man with dark hair and a beard. Below the picture, the name 'John Smith' is displayed in bold black font, followed by the title 'Full Stack Developer'. Underneath the header, there are three navigation links: 'About', 'Bio', and 'Contact', with a thin horizontal line separating them. The 'Bio' link is highlighted with a purple underline. Below the navigation, the word 'Story' is written in bold black font, followed by a block of placeholder text in gray: 'Lorem ipsum, dolor sit amet consectetur adipisicing elit. Quod quis eius eos soluta doloribus a recusandae cum distinctio? Adipisci esse recusandae laborum eligendi voluptas. Eos, impedit dolores! Fugiat, sed unde.'

Private < > codepen.io Stan + Follow

### Interactive Profile Card using tailwind css

HTML

```
1 <div class="min-h-screen bg-gray-100 pt-8">
2   <div class="card mx-auto max-w-sm shadow-md">
3     <!-- header -->
4     <header class="text-center pt-12 pb-8 px-4 bg-yellow-400 rounded-t-lg">
5       <img class="block h-auto mx-auto rounded-full w-20 h-20 mb-3" alt="Profile picture of John Smith" data-tailwind="rounded-full w-20 h-20 mb-3"/>
```

CSS

```
1 @import url("https://fonts.googleapis.com/css?family=Inter:wght@400;600;700&display=swap");
2
3 const tabNavButtons =
4   document.querySelectorAll(".tab-nav button");
5 const marker =
6   document.querySelector(".tab-nav-marker");
7
8 function setMarker() {
9   const activeTab = document.querySelector(".tab.nav-active");
10  if (activeTab) {
11    marker.classList.add("marker-active");
12    activeTab.classList.add("tab-active");
13  }
14}
```

JS (Babel)

```
1 console.clear();
2
3 const tabNavButtons =
4   document.querySelectorAll(".tab-nav button");
5 const marker =
6   document.querySelector(".tab-nav-marker");
7
8 function setMarker() {
9   const activeTab = document.querySelector(".tab.nav-active");
10  if (activeTab) {
11    marker.classList.add("marker-active");
12    activeTab.classList.add("tab-active");
13  }
14}
```

John Smith  
Full Stack Developer

About Bio Contact

**Superpower**  
Writing clean and optimized code

**Tools**  
VS Code, Figma, Github, Netlify

Console Assets Comments ⌘ Keys Fork Embed Export Share

# HWo & GitHub Classroom

Expressing yourself in JSON.

# What's next?

- Check out HWo.
- See you in class on Thursday!