CMPSC 302
WEB DEVELOPMENT

Dig, if you will, some pictures

WE DID THIS ON MONDAY

SOMEONE ELSE'S MACHINE

TODAY WE SCRIPT THIS

Therealkevin

Thinking object-ively

- This communication will require *objects*
  - That is: data "packaged" to travel from one place to another
  - We only get one packet to send, so we need to put it all together
- We also want to script the behavior of our chat
  - As it currently exists on CatChat, how would you define its "workflow?"
    - That is, how does it actually work?

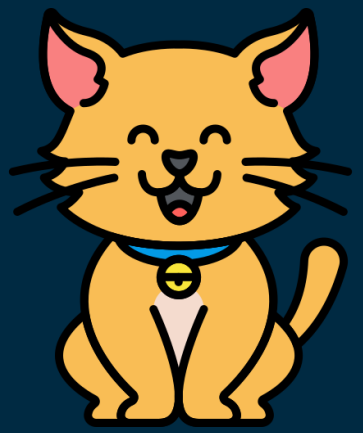
Thinking object-ively

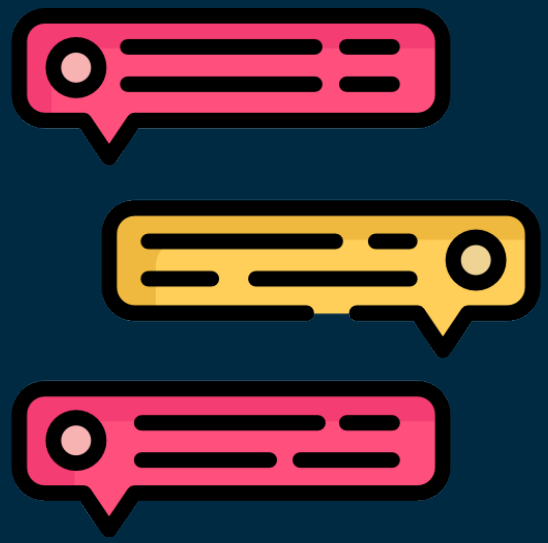
- We might want to start by creating a larger chat *object* that allows us to accomplish all of that
  - This precedes the visual layout, but *affects* it
    - We'll do layout connection on Monday
- Today, we're focusing on making our chat *functional* in the ways you've defined

Thinking object-ively

This whole thing is a “chat” object

Chat starts





Chat ends

Init

Send

Recieve

Send

Recieve

Send

Close

Chat is also the French word for “cat,” so there’s the joke

Serialization


```
var chat = {
  socket: new WebSocket(host),
  init: () => {
        // Chat startup functionality
  },
  send: (message, type) =>
    let msg = {
      user: chat.name,
        text: message,
        type: type
    }
}
```

Serialization

Cerealization