CART 253
Creative Computation 1

Email: l.wilkins@concordia.ca
Office Hours: Tuesday 12-1
Course Github: https://github.com/LeeRobot/CART253-F-22

What we'll be doing today

- Talk about the final assignment
- Learn about MQTT
- In class assignment

Software Rube Goldberg due week 12

- A rube goldberg machine uses one element to trigger another. Usually they are physical, but we will make a software version! Heres an example of a physical rube goldberg machine.

Software Rube Goldberg due week 12

- The goal of this project is to pass along a variable between programs to create a chain. Every person will get a number from someone else, use this number to create a piece of generative art, then pass along a new number to the next person. You must use MQTT to pass a variable.

Software Rube Goldberg due week 12

- Your piece can be ANYTHING: Audio, Visual, interactive, it can use a camera, it can use external data, particle systems, drawing, physical interactions, etc. Your work should be complex and show the skills we learned throughout the semester and things you've explored on your own.

Software Rube Goldberg due week 12

- Your project must:
  - Use **MQTT** protocol to send/receive messages
  - **Receive** a variable from another person
  - **Use** the variable as a factor in creating a visual, auditory, and/or interactive work.
  - **Change** the variable in some way
  - **Send** the variable successfully to the next person
  - **Successfully** complete this task during critique

Software Rube Goldberg due week 12

- Grading criteria:
  - Successfully receiving / changing / using / sending your variables
  - Using 1 external library for p5js in a meaningful capacity
  - Creativity and application
  - Complexity of code

Software Rube Goldberg due week 12

- Sign up: https://docs.google.com/spreadsheets/d/1CKbdBV9WpNAnygMrTLB8C4OHMg2SJD9eVuOk-nyH7v8/edit#gid=0

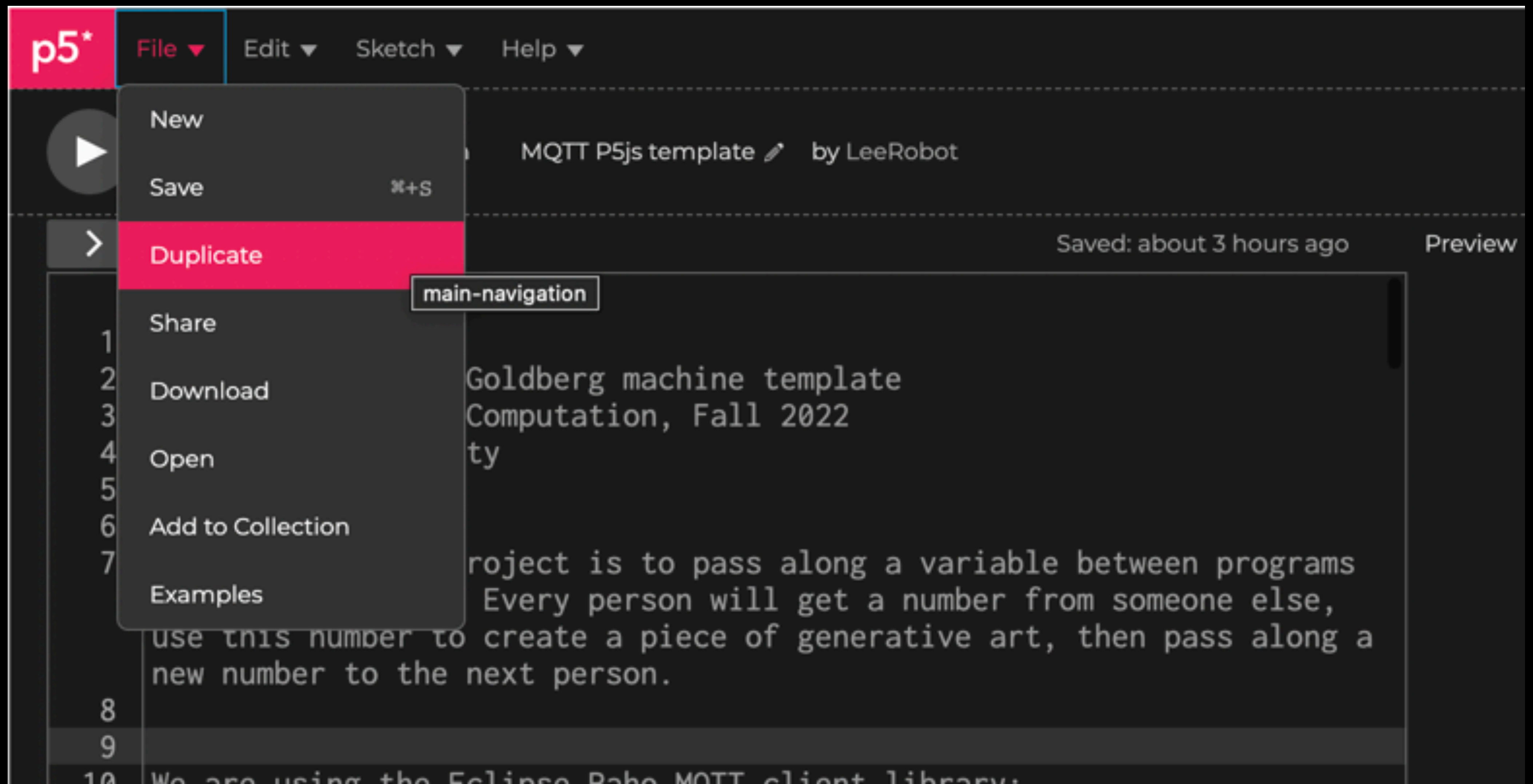
  Lee should be first, Enric should be last.

Template

- Find the template HERE https://editor.p5js.org/LeeRobot/sketches/RXA5yGasr

- Duplicate the template and save it!

- You can use the browser for this assignment, bu tyou can also use Atom if you want.



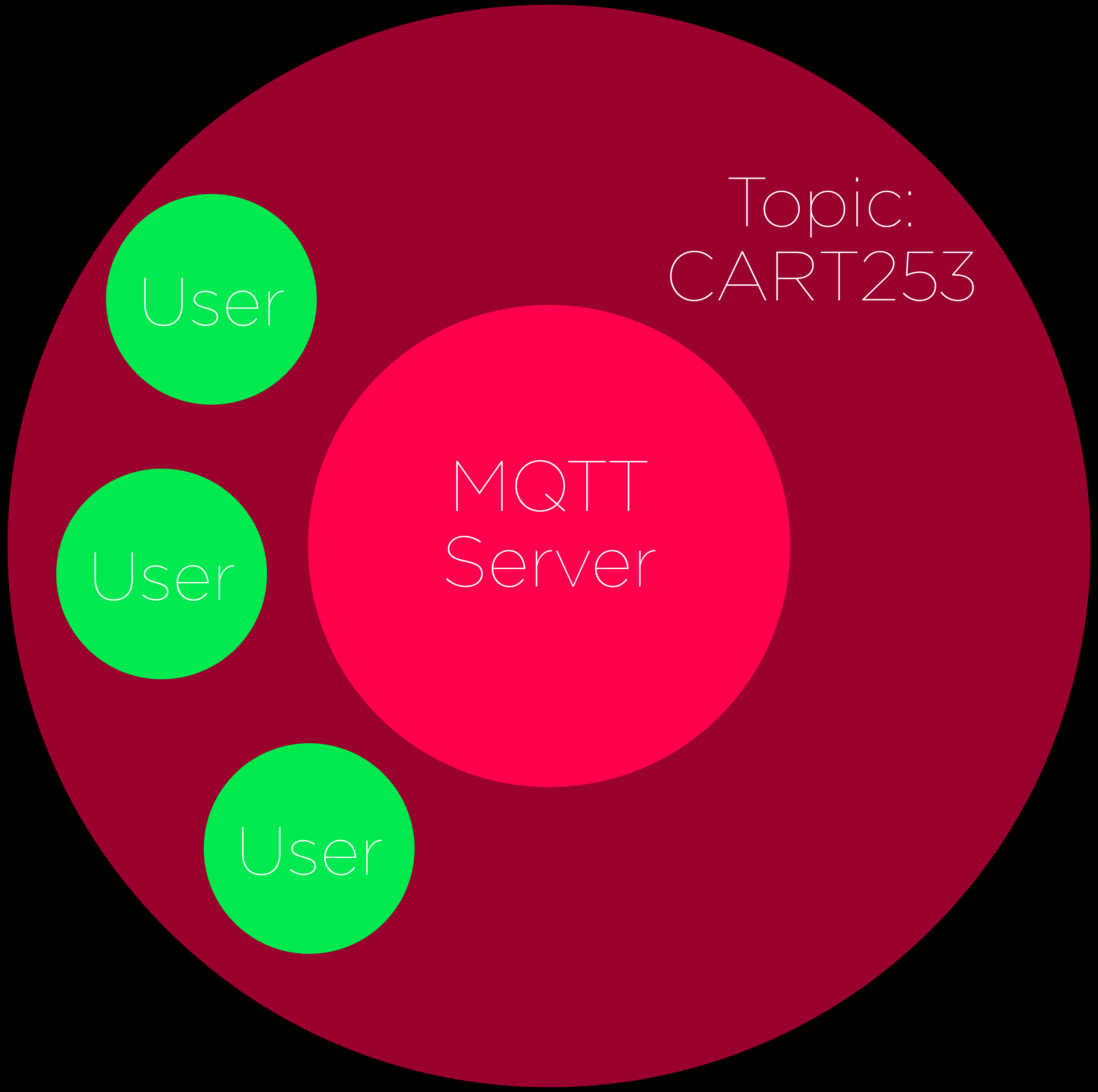
```
Goldberg machine template
Computation, Fall 2022
ty

roject is to pass along a variable between programs
 Every person will get a number from someone else,
use this number to create a piece of generative art, then pass along a
new number to the next person.

We are using the Eclipse Paho MQTT client library:
```

MQTT is a protocol for sending and receiving information over the internet. Users can “subscribe” to “topics” to transfer information.

https://editor.p5js.org/LeeRobot/sketches/umeM7MqWz Here is our basic template for using MQTT with P5JS

MQTT

The MQTT server broadcasts a topic. Users can tune into the topic and view and send messages. By subscribing to a topic, you can see every message any user sends to that topic.



MQTT

We are using a service called Shiftr to do this, this is a public server so everyone can see what you post.

https://www.shiftr.io/try/

shiftr.io

```
let broker = {
  hostname: 'public.cloud.shiftr.io',
  port: 443
};
let client;
let creds = {
  clientID: 'p5Client',
  userName: 'public',
  password: 'public'
}

[..]

function MQTTsetup(){
  client = new Paho.MQTT.Client(broker.hostname, Number(broker.port), creds.clientID);
  client.onConnectionLost = onConnectionLost;
  client.onMessageArrived = onMessageArrived;
  client.connect({
        onSuccess: onConnect,
    userName: creds.userName, // username
    password: creds.password, // password
    useSSL: true
  });
}
```

Connect

```
function sendMQTTMessage(msg) {
    message = new Paho.MQTT.Message(String(msg));
    message.destinationName = topic;
    client.send(message);
}

[...]

 sendMQTTMessage(10);
```

** To send a message, it should be forced to be a "string" of characters

Send

```
let topic = 'CART253';

function onMessageArrived(message) {
  print(message.payloadString);
}
```

This is the topic we are subscribed to

This is a function that happens any time someone sends a message to CART253

This is printing the message

Receive

```
let topic = 'CART253';

function onMessageArrived(message) {
   console.log(message.payloadString);

  if(int(message.payloadString) == 10){
    console.log("yup");
  }
}
```

This is the topic we are subscribed to

This is a function that happens any time someone sends a message to CART253

This is printing the message

** To compare numbers you'll need to make it back into an integer

Receive

```
let myName = "lee"; // Who are you? Make sure it matches the previous person's variable!
let nextName = "KM"; // Who is next on the list? Make sure it matches the next person's variable!
let dataToSend;  // Variable to hold the data to send to the next person on the list
```

Setting up your data

In our template we have some code to facilitate the rube goldberg machine. There are 3 variables to hold your name, and the next person on the list, and data to send.

On our sendMQTTmessage() function, we create a message made out of these 3 variables of information and send it to the topic

In our onMesageArrived() function, we break apart that message and see if its for us, or for someone else.

Setting up your data

```
function sendMQTTMessage(msg) {
     message = new Paho.MQTT.Message(myName + "/" +
nextName+"/"+msg);
// My name + Next name + data separated by /
     message.destinationName = topic;
     console.log("Message Sent!");
     client.send(message);
}
```

You must pass the data you want to send through the function.

This adds all the data points together and separates them by by a / (for the receive function)

This actually sends the message once its all put together

In our template

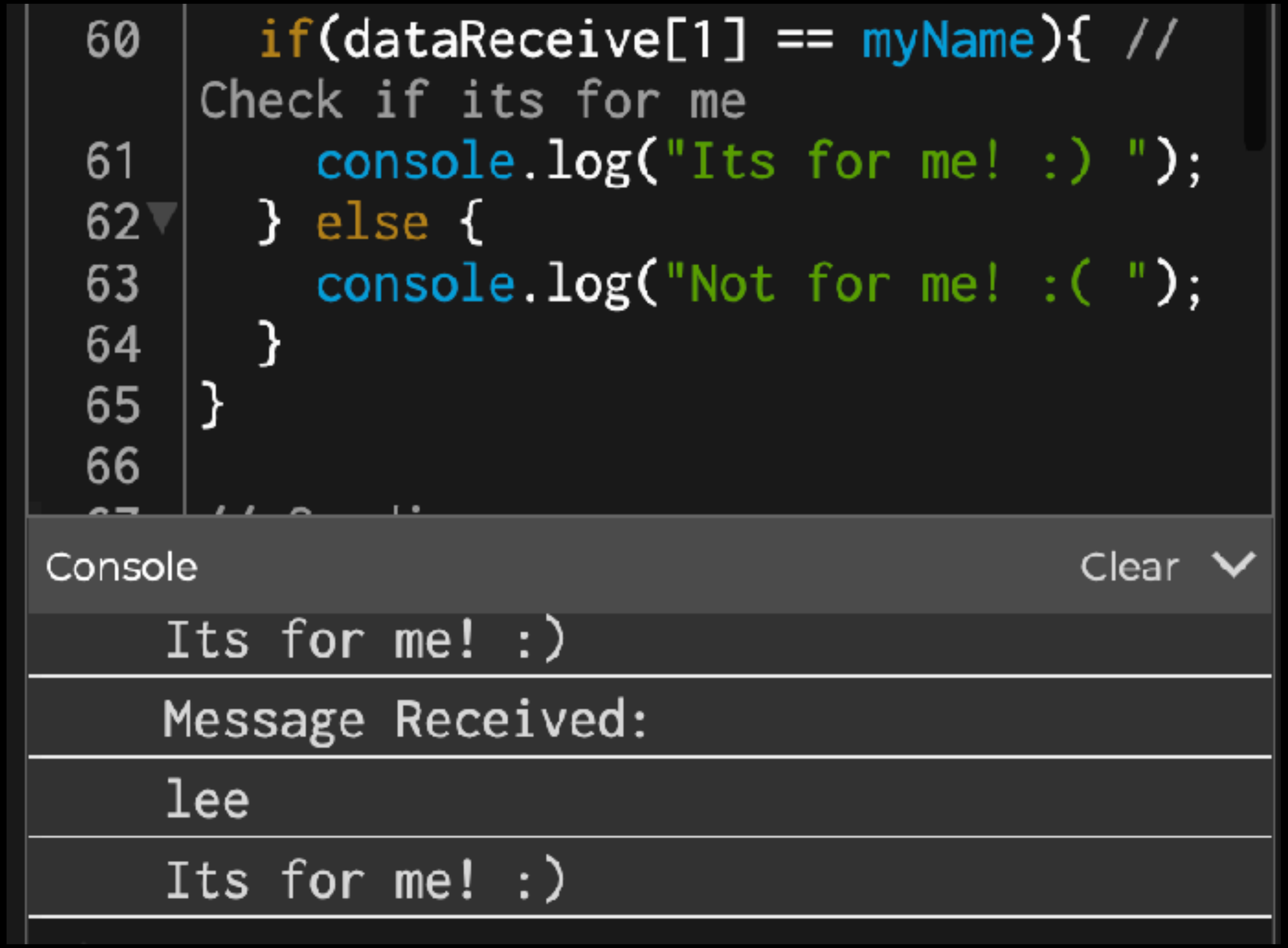
```
function onMessageArrived(message) {
  let dataReceive = split(trim(message.payloadString), "/");
  console.log("Message Received:");
  console.log(String(dataReceive[1]));
// 0 is who its from
// 1 is who its for
// 2 is the data
  if(dataReceive[1] == myName){ // Check if its for me
    console.log("Its for me! :) ");
  } else {
    console.log("Not for me! :( ");
  }
}
```

This splits the incoming message into an array divided by “/“. Each word separated by a / is its own element in the array.

This checks if the second item in the array is the same as the name you entered above.

In our template

```
if(dataReceive[1] == myName){ // Check if its for me
  console.log("Its for me! :) ");
} else {
  console.log("Not for me! :( ");
}
```



```
  if(dataReceive[1] == myName){ //
Check if its for me
    console.log("Its for me! :) ");
  } else {
    console.log("Not for me! :( ");
  }
}
```

Console Clear

```
Its for me! :)
Message Received:
lee
Its for me! :)
```

In our template

```
// Callback functions

function onConnect() {
  client.subscribe(topic);
  console.log("connected");
  // is working
}
function onConnectionLost(response) {
  if (response.errorCode !== 0) {
    // If it stops working
  }
}
```

This happens when the client connects.
This happens when the connection is lost

Functions

To test, you can open two browsers and send messages.
You can set the “myName” and “nextName” variables to create a chain locally

p5*
File
Edit
Sketch
Help
English
Hello, LeeRobot!
Auto-refresh
MQTT P5js template
by LeeRobot
sketch.js
Saved: about 3 hours ago
Preview
Console
Clear
MQTTtestingButtonsLEDs copy

```
*/

// Rube Goldberg setup, update with your own info!
let myName = "lee"; // Who are you? Make sure it matches the previous person's variable!
let nextName = "KM"; // Who is next on the list? Make sure it matches the next person's variable!
let dataToSend;  // Variable to hold the data to send to the next person on the list


// MQTT client details. We are using a public server called shiftr.io. Don't change this.
let broker = {
  hostname: 'public.cloud.shiftr.io'
```

```
connected
Message Received:
lee
Its for me! :)
```

```
update with your own info!
let myName = "KM"; // Who are you? Make sure it matches the previous person's variable!
let nextName = "lee"; // Who is next on the list? Make sure it matches the next person's variable!
let dataToSend;  // Variable to hold the data to send to the next person on the list


// MQTT client details. We are using a public server called shiftr.io. Don't change this.
let broker = {
  hostname: 'public.cloud.shiftr.io',
  port: 443
```

```
Message Sent!
Message Received:
lee
Not for me! :(
```

Set up an MQTT template that sends and receives data from either another browser window or someone else in the class from the CART253 topic **using the template provided**.

When you click anywhere on the canvas, your program should send an MQTT message to someone else that draws a target in the same X and Y position on their canvas as sender clicked. Your program should be able to receive clicks and draw targets as well.

Your targets should be randomly generated color, size, and number of circles (see last assignment).

When you receive a message, draw a green circle on the bottom left **IF its for you**, draw a red circle if it **ISN"T** for you. Draw a blue circle on the right when you send a message.

In Class Assignment