Project 1: Network Disco Lights - HTML5/Javascript

Base

- Create a file called `disco.html`

```
<script type="module">
const WS_URL = "ws://localhost:9800/test1.ws"

function updateScreenColor(r,g,b) {
    function randomByte() {return Math.floor(Math.random()*255)}
    document.body.style = `background-color: rgb(${r||randomByte()},${g||randomByte()},${b||randomByte()});`
}

</script>
```

- Open `disco.html` in a web browser
- Replace `localhost` with URL your teacher gives you
- Run the program after each addition by refreshing your browser `ctrl + r`

Random Screen Color

```
+
+window.addEventListener('keydown', (event)=>{
+    updateScreenColor()
+})

 </script>
```

Press any key on your keyboard, your screen should change color.

Listen to Network

```
+const socket = new WebSocket(WS_URL)
+window.socket = socket
+socket.addEventListener('message', (event)=>{
+            updateScreenColor()
+})

 </script>
```

When a network message is revived, your screen should change color.

Protocol Version 1: Computer Id

Change "1" to be a unique Identifier (ID) for your computer.
Your teacher will give you a number.

```
 socket.addEventListener('message', (event)=>{
+        console.log("received", event.data)
+        const id = event.data
+        if (id == "1") {
             updateScreenColor()
+        }
 })
```

When your number is sent over a network, your screen should change color.

Press F12 to enable developer tools 'console'.

```
socket.send("1")
```

Change the color of your neighbors screen with their `id`

Protocol Version 2: Color

```
 socket.addEventListener('message', (event)=>{
         console.log("received", event.data)
-        ~~const id = event.data~~
+        const [id, r, g, b] = event.data.split(":")
         if (id == "1") {
-            ~~updateScreenColor()~~
+            updateScreenColor(r,g,b)
        }
 })
```

In console

```
socket.send("1:255:0:0")
```

Change the color of your neighbors screen to your own color with their `id`

Protocol Version 3: Multiple Computer Id's

```
 socket.addEventListener('message', (event)=>{
     console.log("received", event.data)
+    const messages = event.data.split("\n")
+    for (let message of messages) {
-        ~~const [id, r, g, b] = event.data.split(":")~~
+        const [id, r, g, b] = message.split(":")
         if (id == "1") {
             updateScreenColor(r,g,b)
         }
+    }
 })
```

In console

```
socket.send(`
1:255:0:0
2:0:255:0
3:0:0:255
`)
```

With a single multiline message; Change the color of your two neighbors screens to two different colors with their `id` s
Hint: You will need `shift + enter` for new lines and backticks (next to `1` on the keyboard)

Calculate BPM

- Calculate the Beats Per Minuet (BPM) of a piece of music
- Count/tally the number of beats over 60 seconds
- Disco Time

Extra

- Join another channel with a small group - replace `test1` for different channels. e.g. `my_cool_channel`

Project 2: Remote Control (Chromecast clone)

The example below will help you build your basic 'chromecast' like system with a `display` and `control` machines.
Your going to have to think about how these examplem peices can be combined.

Reminder from Project1

```
const WS_URL = "wss://YOUR_SERVER_URL/YOUR_CHANNEL_NAME.ws"
const socket = new WebSocket(WS_URL)
window.socket = socket
socket.addEventListener('message', (event)=>{
})
```

`display.html`

Display - image, text, video (mp4, youtube), redirect-webpage (frame?) (need snippets)

```
document.body.style = "margin:0; background-color:black;"  // remove the default html border
```

```
function clear() {
  document.body.innerHTML = ``
}
```

```
function iframe(url) {
  document.body.innerHTML = `<iframe style="width: 100%; height: 100%;" src="${url}" frameborder="0" allowfullscreen></iframe>`
}
iframe("https://canterbury.ac.uk/")
```

```
function youtube(video_id, start_seconds=0) {
  iframe(`https://www.youtube.com/embed/${video_id}?&autoplay=1&start=${start_seconds}`)
}
youtube("0jjeOYMjmDU")
```

```
function image(image_url) {
  document.body.innerHTML = `<img style="width: 100%; height: 100%; object-fit: cover;" src="${image_url}">`
}
image("https://upload.wikimedia.org/wikipedia/commons/6/60/Augustine_Abbey.jpg")
// todo: animate, fade, blur
```

```
function media(url) {
  document.body.innerHTML = `<video controls autoplay style="width: 100%; height: 100%; object-fit: contain;"><source src="${ur
}
media("https://file-examples-com.github.io/uploads/2017/04/file_example_MP4_1280_10MG.mp4")
// or mp3/ogg
```

consider `mute` and `loop` and `controls`

```
function speak(text) {
  window.speechSynthesis.speak(new SpeechSynthesisUtterance(text));
}
speak('Hello World');
```

```
function text(text) {
  document.body.innerHTML = `<p>${text}</p>`
}
text('Hello World');
// TODO: set font size and color and font? center? wrap?
// Animation? scroll in?
```

```
const renderer_functions = {clear, iframe, youtube, image, media, speak, text}
// ...
  const function_name, payload = event.data.split("---")
  renderer_functions[function_name](payload)
// ...
  // socket.send(`youtube---A67ZkAd1wmI`)
```

Explain use `---` as split character. Don't use `:` because this is a legal character in urls

`control.html`

Create a separate
Remote control - with buttons and text box's. Slider? color picker?

Demo of a button

Paste this into a browser javascript console for a simple button demo.

```
const button = document.createElement("button")
button.textContent = "Console Hello"
button.addEventListener("click", ()=>{
    console.log("hello")
})
document.body.appendChild(button)
```

HTML template

Save this as a `.html` file and load it in your browser.

```
<h1>Test</h1>

<button id="button1">Test Button 1</button>

<input type="text" id="text1"/>
<button id="button2">Test Button 2 - from textbox</button>

<label for="points">Points (between 0 and 10):</label>
<input type="range" id="points" name="points" min="0" max="10">

<script type="module">
    // TODO: const socket = new WebSocket(WS_URL)

    function do_thing(text) {
        console.log(text)
        // TODO: socket.send(`MY MESSAGE ${text}`)
    }

    document.getElementById("button1").addEventListener("click", ()=>{
        do_thing('hello1')
    })

    document.getElementById("button2").addEventListener("click", ()=>{
        const text = document.getElementById("text1").value
        do_thing(`Text in the box is ${text}`)
    })
</script>
```

Advanced: Mouse-move events