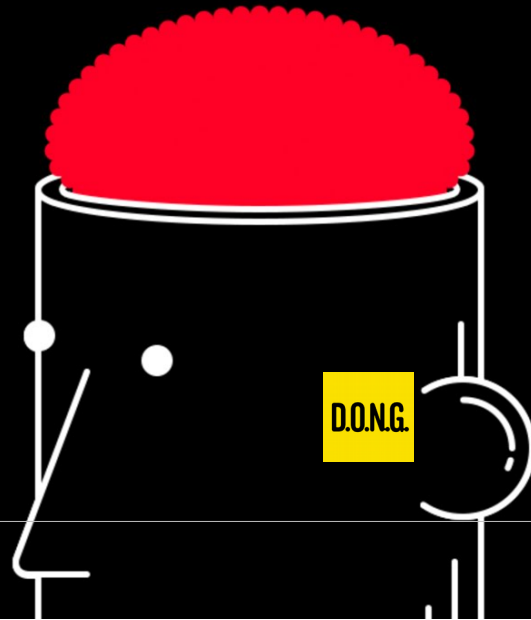


Contemporary Web Development

Lesson 8



Yesterday's exercise –
Sending numbers efficiently

Yesterday's exercise –
Sending numbers efficiently

Populating the ArrayBuffer

0	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	
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	?

```
let buffer = new ArrayBuffer(4);  
let id = new Uint16Array(buffer);  
id[0] = 65535; // Maximum value 2^16 - 1
```

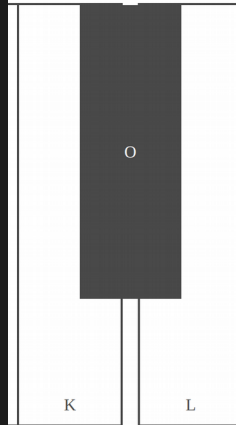
String to UInt18Array (Or UInt16Array for "special" characters)

```
let buffer = new ArrayBuffer(3);  
let chars = new Uint8Array(buffer);  
let string = "SMH";  
chars[0] = string.charCodeAt(0);  
chars[1] = string.charCodeAt(1);  
chars[2] = string.charCodeAt(2);  
// Of course this could be done in a loop
```

Having the Collab Synth
support up to 65535 cilents

Debugging Websocket Data*

* But will not see the contents of
binary messages



The screenshot shows the Network tab of a browser's developer tools. The 'WS' (Websocket) filter is selected. A list of websocket messages is displayed, including frames like 7Ps, 7Ss, 1Pa, 1Sa, 1Ps, 1Pd, 1Ss, 1Sd, 1Pf, 1Sf, 1Ph, 1Sh, 1Pg, 1Sg, 1Pp, and 1Sp. The '1Sf' message is highlighted in blue. The 'Data' column shows the content of the selected message, which is '1 Sf'. The status bar at the bottom indicates '1 requests | 0 B transferred'.

Summary of Client modules

- **InputHandler** – receives input events, then triggers application events.
- **SocketClient** – Triggers socket events, can send data to the server.
- **Synth** - Controls the audio.
- **Piano** – Controls the HTML piano display.
- **Index** – connects everything together. Redirects HTML events and application events to the correct module.
- **State** – storage for the application state, each module has its own storage object.

Small Exercises

- 1) Change the color that is assigned to your own synth keystrokes to something else.
- 2) Change the synth to sine wave instead of square wave.
- 3) Disable mouse play completely.

Put your text on the screen challenge

Bigger Exercises

- 1) Have the Input module count and store in its own state, how many times any key was clicked using the mouse. Log accumulated number to the console every time the mouse clicks the keys.
- 2) Add a button that causes every connected client's screen to finitiates a "Kick drum" as in [here](#).
- 3) Add a volume control slider.

Debugging Websocket Binary Data*

* But can't install it in the IT
Classroom, you can always
console.log

The image shows a Wireshark network traffic capture window. The title bar reads "Activities Wireshark". The menu bar includes "File", "Edit", "View", "Go", "Capture", "Analyze", "Statistics", "Telephony", "Wireless", "Tools", and "Help". The toolbar contains various icons for file operations, capture, and analysis. The main pane is titled "websocket" and displays a list of captured packets. The table below shows the packet list:

No.	Time	Source	Destination	Protocol	Length	Info
194	8.763033083	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
198	8.793499408	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
202	8.823583234	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
206	8.853265239	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
210	8.884544377	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
214	8.915597402	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
218	8.945466792	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
222	8.976073203	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
226	9.006869963	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
230	9.028087309	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
234	9.758816159	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
238	9.773233740	127.0.0.1	127.0.0.1	WebSoc...	74	WebSocket
240	9.774095550	127.0.0.1	127.0.0.1	WebSoc...	70	WebSocket

Packet 230 is selected. The packet details pane shows:

- Frame 230: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
- Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)
- Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
- Transmission Control Protocol, Src Port: 41794, Dst Port: 3000, Seq: 770, Ack: 130, Len: 74, Window: 0
- WebSocket
- Data (2 bytes)
- Data: 5361
- [Length: 2]

The packet bytes pane shows the raw data: 0000 53 61. The hex value 53 corresponds to the ASCII character 'S' and 61 to 'a'.