

Binary Clock

1 Introduction

A binary clock displays sexagesimal time using a binary format - see Figure 1. The bottom row in each column represents 1, with each row above representing higher powers of two, up to 8. To read the time, you add the values on each column. The first two columns represent the hour, the next two represent the minutes and the last two represent the seconds.

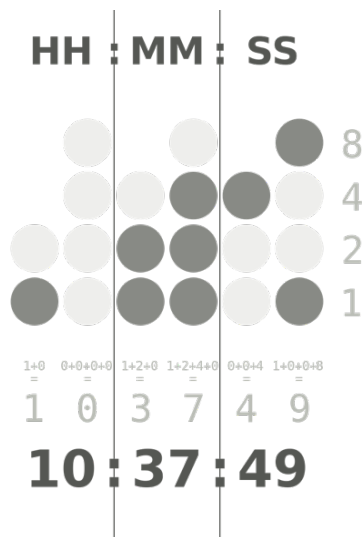


Figure 1: Binary Clock Model

In this homework you will create a binary clock using HTML and JavaScript - see Figure 2

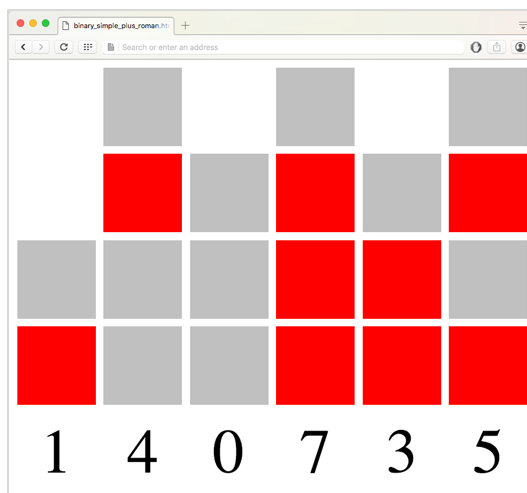


Figure 2: HTML/CSS Binary Clock

2 Questions

Resources

The link to starter code for the homework can be found on <http://onexi.org/homeworks.html>. Please follow the guidelines discussed in class for submission.

2.1

Modify the starter code so that the `timer` function is called `onload` (lines 31-to-32). Verify your code is writing the time to the console.

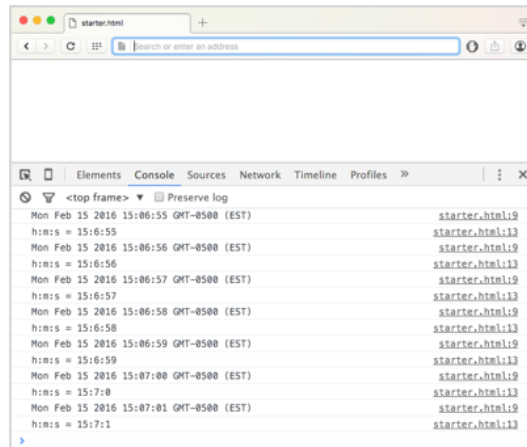


Figure 3: Output using Debugger

2.2

Write a binary clock grid using HTML elements to hold binary clock positions (lines 40-to-46). Use CSS to create the squares. Your completed grid should look like - see Figure 5. You can modify the style on line 36 if needed.

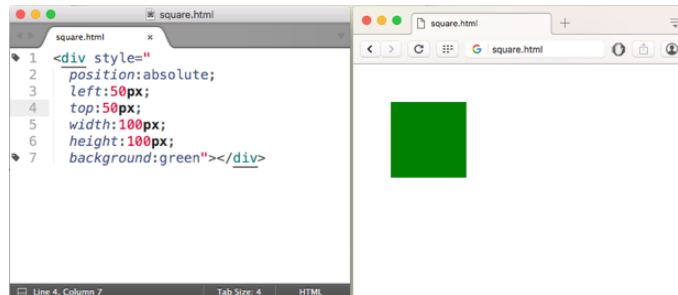


Figure 4: Example of Grid Element

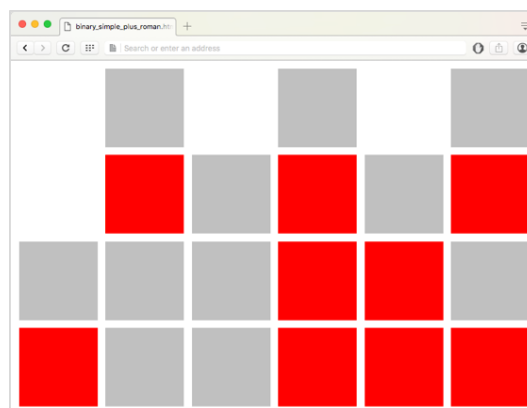


Figure 5: Completed Grid

2.3

Write JavaScript in the `setBinary` function (lines 26-to-29) to update the HTML clock positions.

2.4

Add an additional row of elements to display traditional time, e.g. 14:07:35. The dashed shape boundaries are not needed - see Figure 6.



Figure 6: Roman Numeral Row

2.5

Write JavaScript in the `setRoman` function (lines 21-to-24) to write roman time to the new row.



Figure 7: Populated Numerals

Note

Please note that you must implement and use `exercise.roman()` in function `setRoman()`. Similarly, you must implement and use `exercise.binary()` in function `setBinary()`. Both functions are in the `exercise.js` file.

3 Appendix

run.html

```
1 <!DOCTYPE html>
2 <html>
3 <script src="exercise.js"></script>
4 <script>
5
6 var timer = function() {
7     // get the current date & time
8     var time = new Date();
9     console.log(time);
10
11     // show roman time, using 6 columns
12     setRoman(time);
13
14     // show binary time
15     setBinary(time);
16
17     // update every 1 second
18     setTimeout(timer, 1000);
19 };
20
21 var setRoman = function(time){
22     // set roman time
23     // you must implement exercise.roman()
24 };
25
26 var setBinary = function(time){
27     // set roman time
28     // you must implement exercise.binary()
29 };
30
31 // start timer when page is loaded
32 // your code here
33
34 </script>
35
36 <style> div {position:absolute; width: 100px; height: 100px;
37 background: silver; text-align: center;vertical-align: middle;
38 line-height: 100px;font-size: 5em;} </style>
39
40 <!--
41
42     Write HTML elements to hold your binary clock positions
43
44     There are 6 columns in a binary clock
45
46 -->
47
48 </html>
```

exercise.js

```
1 var exercise = {};
2
3
4 exercise.roman = function(time) {
5
6     //-----
7     // YOUR CODE
8     // Return an object with roman time.
9     // Time is reported using 6 columns.
10    //
11    // For example, for 05:13:47 PM
12    // {
13    //     hour_col1 : 1,
14    //     hour_col2 : 7,
15    //     min_col1  : 1,
16    //     min_col2  : 3,
17    //     sec_col1  : 4,
18    //     sec_col2  : 7
19    // };
20    //
21    //-----
22
23 };
24
25 exercise.binary = function(time, col) {
26
27     //-----
28     // YOUR CODE
29     // Return an object with
30     // the binary clock values
31     // for the given column
32     //
33     // For example, for time 05:13:47 PM, and column hour_col2
34     // var binary = {
35     //     position8 : 'off',
36     //     position4 : 'on',
37     //     position2 : 'on',
38     //     position1 : 'on',
39     // };
40     //
41     //-----
42
43 };
```