

File Edit Selection View Go Run ...

task2.html

```
orphan tech > task2.html > html > body > script
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <title>Stopwatch with Lap</title>
6    <style>
7      body {
8        font-family: Arial, sans-serif;
9        background: #f1f1f1;
10       color: #fff;
11       display: flex;
12       justify-content: center;
13       align-items: center;
14       height: 100vh;
15     }
16
17     .stopwatch {
18       background: #2b2b3d;
19       padding: 25px;
20       border-radius: 10px;
21       width: 300px;
22       text-align: center;
23       box-shadow: 0 10px 25px rgba(0,0,0,0.4);
24     }
25
26     .time {
27       font-size: 2.5em;
28       margin-bottom: 20px;
29       letter-spacing: 2px;
30     }
31
32     .buttons {
33       display: flex;
34       justify-content: space-between;
35       margin-bottom: 15px;
36     }
37
38     button {
39       flex: 1;
40       margin: 0 5px;
41       padding: 10px;
42       font-size: 0.9em;
43     }
44   </style>
45   <script>
46     let stopwatch = document.querySelector('.stopwatch');
47     let timeElement = stopwatch.querySelector('.time');
48     let startButton = stopwatch.querySelector('button');
49     let lapButton = stopwatch.querySelector('button');
50     let resetButton = stopwatch.querySelector('button');
51
52     let isRunning = false;
53     let lapCount = 0;
54
55     startButton.addEventListener('click', () => {
56       if (!isRunning) {
57         start();
58       } else {
59         stop();
60       }
61     });
62
63     lapButton.addEventListener('click', () => {
64       if (isRunning) {
65         lap();
66       }
67     });
68
69     resetButton.addEventListener('click', () => {
70       reset();
71     });
72
73     function start() {
74       isRunning = true;
75       let intervalId = setInterval(() => {
76         let time = new Date().toLocaleTimeString();
77         timeElement.textContent = time;
78       }, 1000);
79     }
80
81     function stop() {
82       isRunning = false;
83       clearInterval(intervalId);
84     }
85
86     function lap() {
87       if (isRunning) {
88         let time = new Date().toLocaleTimeString();
89         console.log(`Lap ${lapCount}: ${time}`);
90         lapCount++;
91       }
92     }
93
94     function reset() {
95       isRunning = false;
96       stopwatch.innerHTML = '';
97       stopwatch.innerHTML = `
98         <div class="stopwatch">
99           <div class="time">00:00:00</div>
100          <div class="buttons">
101            <button>Start</button>
102            <button>Lap</button>
103            <button>Reset</button>
104          </div>
105        </div>
106      `;
107     }
108   </script>
```

Ln 164, Col 25 Spaces: 2 UTF-8 CRLF ⌂ HTML ⌂ Port: 3000 ⌂ Go Live ⌂ Prettier

File Edit Selection View Go Run ...

task2.html X

orphion tech > task2.html > html > head > style > buttons

```
2  <html lang="en">
3  <head>
4    <style>
5
6      button {
7        border: none;
8        border-radius: 5px;
9        cursor: pointer;
10     }
11
12     button:disabled {
13       opacity: 0.5;
14       cursor: not-allowed;
15     }
16
17     .start { background: #4caf50; color: #fff; }
18     .pause { background: #ff9800; color: #fff; }
19     .reset { background: #f44336; color: #fff; }
20     .lap   { background: #2196f3; color: #fff; }
21
22     ul {
23       list-style: none;
24       padding: 0;
25       max-height: 150px;
26       overflow-y: auto;
27       text-align: left;
28     }
29
30     li {
31       padding: 5px;
32       border-bottom: 1px solid #444;
33       font-size: 0.9em;
34     }
35   </style>
36 </head>
37 <body>
38
39   <div class="stopwatch">
40     <div class="time" id="display">00:00:00.000</div>
41
42     <div class="buttons">
43       <button class="start" id="startBtn">Start</button>
44       <button class="pause" id="pauseBtn" disabled>Pause</button>
45     </div>
46   </div>
47
48   <ul>
49     <li>Stopwatch</li>
50     <li>Reset</li>
51     <li>Lap Counter</li>
52   </ul>
53
54   <script src="script.js"></script>
55
56 </body>
57 </html>
```

In 35, Col 27 Spaces: 2 UTF-8 CRLF ↻ HTML 🌐 Port: 3000 ⚡ Go Live ✨ Prettier

The screenshot shows a browser-based code editor with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Toolbar:** Back, Forward, Refresh, Coding, Copy, Paste, Undo, Redo.
- Code Area:** The file `task2.html` is open. The code is a stopwatch application with HTML, CSS, and JavaScript. It includes a start button, a pause button, a reset button, and a lap button. It also tracks laps and displays time in hours, minutes, and seconds.
- Right Panel:** Shows a sidebar with code snippets and a preview area.
- Status Bar:** Line 35, Col 27, Spaces: 2, UTF-8, CRLF, HTML, Port: 3000, Go Live, Prettier.

```
<html lang="en">
<body>
<div class="stopwatch">
    <div class="buttons">
        <button class="start" id="startBtn">Start</button>
        <button class="pause" id="pauseBtn" disabled>Pause</button>
    </div>
    <div class="buttons">
        <button class="lap" id="lapBtn" disabled>Lap</button>
        <button class="reset" id="resetBtn" disabled>Reset</button>
    </div>
    <ul id="laps"></ul>
</div>
<script>
let startTime = 0;
let elapsedTime = 0;
let timerInterval;
let running = false;
let lapCount = 1;

const display = document.getElementById("display");
const startBtn = document.getElementById("startBtn");
const pauseBtn = document.getElementById("pauseBtn");
const resetBtn = document.getElementById("resetBtn");
const lapBtn = document.getElementById("lapBtn");
const laps = document.getElementById("laps");

function formatTime(time) {
    const ms = time % 1000;
    const totalSeconds = Math.floor(time / 1000);
    const seconds = totalSeconds % 60;
    const minutes = Math.floor(totalSeconds / 60) % 60;
    const hours = Math.floor(totalSeconds / 3600);

    return (
        String(hours).padStart(2, "0") + ":" +
        String(minutes).padStart(2, "0") + ":" +
        String(seconds).padStart(2, "0") + "." +
        String(ms).padStart(3, "0")
    );
}

startBtn.addEventListener("click", () => {
    if (!running) {
        startTime = Date.now();
        running = true;
        pauseBtn.disabled = false;
        resetBtn.disabled = false;
        lapBtn.disabled = false;
        timerInterval = setInterval(() => {
            elapsedTime = Date.now() - startTime;
            display.textContent = formatTime(elapsedTime);
        }, 100);
    } else {
        clearInterval(timerInterval);
        running = false;
        pauseBtn.disabled = true;
        resetBtn.disabled = false;
        lapBtn.disabled = true;
    }
});

resetBtn.addEventListener("click", () => {
    startTime = 0;
    elapsedTime = 0;
    display.textContent = "00:00:00.000";
    pauseBtn.disabled = true;
    resetBtn.disabled = true;
    lapBtn.disabled = true;
    clearInterval(timerInterval);
    timerInterval = null;
    running = false;
});

lapBtn.addEventListener("click", () => {
    lapCount++;
    laps.innerHTML += `<li>${formatTime(elapsedTime)}</li>`;
});</script>
```

A screenshot of a code editor window titled "task2.html". The window includes a toolbar with icons for File, Edit, Selection, View, Go, Run, and others. The main area shows the following JavaScript code:

```
orphan tech > task2.html > html > head > style > buttons
  2   <html lang="en">
  3     <body>
  4       <script>
  5         function formatTime(time) {
  6           return String(ms).padStart(3, "0");
  7         }
  8
  9         function updateDisplay() {
 10           const currentTime = Date.now() - startTime + elapsedTime;
 11           display.textContent = formatTime(currentTime);
 12         }
 13
 14         startBtn.addEventListener("click", () => {
 15           if (!running) {
 16             startTime = Date.now();
 17             timerInterval = setInterval(updateDisplay, 10);
 18             running = true;
 19           }
 20           startBtn.disabled = true;
 21           pauseBtn.disabled = false;
 22           resetBtn.disabled = false;
 23           lapBtn.disabled = false;
 24         });
 25
 26         pauseBtn.addEventListener("click", () => {
 27           if (running) {
 28             clearInterval(timerInterval);
 29             elapsedTime += Date.now() - startTime;
 30             running = false;
 31           }
 32           startBtn.disabled = false;
 33           pauseBtn.disabled = true;
 34         });
 35
 36         resetBtn.addEventListener("click", () => {
 37           clearInterval(timerInterval);
 38           startTime = 0;
 39           elapsedTime = 0;
 40           running = false;
 41         });
 42
 43         display.textContent = formatTime(elapsedTime);
 44       </script>
 45     </body>
 46   </html>
```

The code defines a script block containing functions for formatting time, updating the display, starting, pausing, and resetting a timer. It uses the `Date.now()` method to track time and `setInterval` to update the display every 10ms. Buttons are disabled or enabled based on the state of the timer.

The screenshot shows a browser-based code editor with the following details:

- File Bar:** File, Edit, Selection, View, Go, Run, ...
- Search Bar:** Q Coding
- Toolbar:** Back, Forward, Home, Stop, Refresh, Reload, Minimize, Maximize, Close.
- Left Sidebar:** Includes icons for file operations (New, Open, Save, Find, Replace, Copy, Paste, Undo, Redo), a search bar, and a tree view of the current file structure.
- Code Area:** The main content displays the `task2.html` file. The code is a script for a lap timer application. It includes logic for start, pause, reset, and lap buttons, as well as a display showing time and lap counts.

```
task2.html X
orphan tech > task2.html > html > head > style > buttons
2   <html lang="en">
3     <body>
4       <script>
5         resetBtn.addEventListener("click", () => {
6           const startTime = Date.now();
7           running = false;
8           lapCount = 1;
9
10          display.textContent = "00:00:00.000";
11          laps.innerHTML = "";
12
13          startBtn.disabled = false;
14          pauseBtn.disabled = true;
15          resetBtn.disabled = true;
16          lapBtn.disabled = true;
17        });
18
19        lapBtn.addEventListener("click", () => {
20          if (running) {
21            const lapTime = Date.now() - startTime + elapsedTime;
22            const li = document.createElement("li");
23            li.textContent = `Lap ${lapCount++}: ${formatTime(lapTime)}`;
24            laps.prepend(li);
25          }
26        });
27      </script>
28    </body>
29  </html>
30
```

- Bottom Status Bar:** Line 35, Col 27, Spaces: 2, CRLF, { HTML, CSS, JS, Port: 3000, Go Live, Prettier, and a refresh icon.