```
// Auto-Generated Random Paragraphs for typing
 1
                                                                                                                                                                 CONFIGURE
 2
           const paragraphs = [
 3
                  "Typing fast with all ten fingers is a skill every programmer should master. I
 4
                  "Becoming a fast typist begins with using the correct typing posture. Sit up so that the state of the state o
 5
 6
 7
                                                                                                                                                                 Function with the larg
 8
                  "Learning programming and touch typing go hand in hand. As you practice codiangout (Mahile the median is C
 9
10
                  "The journey to mastering typing involves consistency and patience. Begin by prumeriation (MacCS)
11
                                                                                                                                                                 median is 3.
12
                  "Did you know that the world's fastest typists often use the same strategies a: Install (/install) The most complex fu
13
14
                                                                                                                                                      Contribute (contribute) the
15
                  "Touch typing is not only a skill for speed but also for multitasking. Program
                                                                                                                                                      Blog (/blog)
16
17
                  "Fast typing doesn't mean rushing through code. Instead, it's about maintaining
                                                                                                                                                                 35 warnings
18
19
                  "Typing with ten fingers boosts productivity and reduces stress during coding r
                                                                                                                                                                     2 'const' is availa
20
                                                                                                                                                                          Mozilla JS exte
21
                  "Programming and typing are like two sides of the same coin. When you improve (
22
              ];
                                                                                                                                                                   25 'const' is availa
23
24
          // Selecting DOM elements
                                                                                                                                                                          Mozilla JS exte
           const typingText = document.querySelector(".typing-text p");
25
                                                                                                                                                                   26 'const' is availa
           const inpField = document.querySelector(".wrapper .input-field");
26
           const tryAgainBtn = document.querySelector(".content button");
27
                                                                                                                                                                          Mozilla JS exte
          const timeTag = document.querySelector(".time span b");
28
                                                                                                                                                                   27 'const' is availa
29
           const mistakeTag = document.querySelector(".mistake span");
          const wpmTag = document.querySelector(".wpm span");
const cpmTag = document.querySelector(".cpm span");
30
                                                                                                                                                                         Mozilla JS exte
31
                                                                                                                                                                   28 'const' is availa
32
33
          // Declaring variables for game logic
                                                                                                                                                                         Mozilla JS exte
34
          let timer;
35
          let maxTime = 60;
                                                                                                                                                                   29 'const' is availa
36
           let timeLeft = maxTime;
                                                                                                                                                                          Mozilla JS exte
37
          let charIndex = 0;
38
          let mistakes = 0;
                                                                                                                                                                   30 'const' is availa
39
          let isTyping = false;
                                                                                                                                                                         Mozilla JS exte
40
41
           // Load a random paragraph and display it
                                                                                                                                                                   31 'const' is availa
42
           function loadParagraph() {
                                                                                                                                                                          Mozilla JS exte
              const randomIndex = Math.floor(Math.random() * paragraphs.length);
43
44
              typingText.innerHTML = [...paragraphs[randomIndex]]
                                                                                                                                                                   34 'let' is available
45
                  .map(char => `<span>${char}</span>`)
                                                                                                                                                                         Mozilla JS exte
46
                  .join("");
47
                                                                                                                                                                   35 'let' is available
48
              typingText.firstElementChild.classList.add("active");
49
                                                                                                                                                                         Mozilla JS exte
              // Automatically focus the input field on keydown or click
50
                                                                                                                                                                   36 'let' is available
51
              const focusInput = () => inpField.focus();
              document.addEventListener("keydown", focusInput);
typingText.addEventListener("click", focusInput);
52
                                                                                                                                                                          Mozilla JS exte
53
                                                                                                                                                                   37 'let' is available
54
55
                                                                                                                                                                         Mozilla JS exte
56
           // Typing Logic Function
57
                                                                                                                                                                   38 'let' is available
           function handleTyping() {
              const characters = typingText.querySelectorAll("span");
58
                                                                                                                                                                         Mozilla JS exte
59
              const typedChar = inpField.value[charIndex];
60
                                                                                                                                                                   39 'let' is available
61
              if (charIndex >= characters.length || timeLeft <= 0) return finishGame();</pre>
                                                                                                                                                                         Mozilla JS exte
62
63
              if (!isTyping) startTimer();
                                                                                                                                                                   43 'const' is availa
64
                                                                                                                                                                         Mozilla JS exte
65
              //Allowing user to backspace
66
              if (typedChar == null) {
                                                                                                                                                                   44 'spread operator
                     if (charIndex > 0)
67
                                                                                                                                                                          'esversion: 6').
                            charIndex--; // Move back to the previous character
68
                            if (characters[charIndex].classList.contains("incorrect")) mistakes--
69
                                                                                                                                                                   45 Homplata litaral
                            characters[charIndex].classList.remove("correct", "incorrect", "active")
70
```

https://jshint.com

```
71
             }
72
73
        } else { // Check Character if correct or incorrect
74
             const isCorrect = characters[charIndex].innerText === typedChar;
75
             characters[charIndex].classList.add(isCorrect ? "correct" : "incorrect");
76
             if (!isCorrect) mistakes++;
                                                                                          version 2.13.6
There are 13 function
(https://github.com/jshint/js
77
             charIndex++;
78
79
                                                                                                 Function with the larg
80
      //Update activated character
                                                                                           About (While the median is C
        characters.forEach(span => span.classList.remove("active"));
81
        if (charIndex < characters.length) characters[charIndex].classList.add("active") Largest function has
82
83
                                                                                                 median is 3.
                                                                                           Install (/install)
The most complex fu
84
        updateStats();
85
      }
86
       // Timer function
                                                                                           Contribute (contribute) the
87
      function startTimer() {
                                                                                           Blog (/blog)
88
        isTyping = true;
89
        timer = setInterval(() => {
                                                                                                 35 warnings
90
             if (--timeLeft >= 0) {
91
                 timeTag.innerText = timeLeft;
                                                                                                    2 'const' is availa
                 updateStats(); // Update WPM during countdown
92
                                                                                                       Mozilla JS exte
93
94
                 clearInterval(timer); // Stopping the timer
                                                                                                   25 'const' is availa
95
                 finishGame();
                                                                                                      Mozilla JS exte
96
        }, 1000);
97
                                                                                                   26 'const' is availa
98
                                                                                                       Mozilla JS exte
99
100
      // Update WPM, CPM, and mistakes
                                                                                                   27 'const' is availa
101
      function updateStats() {
102
        const wpm = Math.max(Math.round(((charIndex - mistakes) / 5) / ((maxTime - timeLounder))
                                                                                                      Mozilla JS exte
103
        wpmTag.innerText = wpm;
104
        cpmTag.innerText = charIndex - mistakes;
                                                                                                   28 'const' is availa
105
        mistakeTag.innerText = mistakes;
                                                                                                      Mozilla JS exte
106
                                                                                                   29 'const' is availa
107
108
      // Reset the game
                                                                                                       Mozilla JS exte
109
      function resetGame() {
110
        clearInterval(timer);
                                                                                                   30 'const' is availa
         [timeLeft, charIndex, mistakes, isTyping] = [maxTime, 0, 0, false];
111
                                                                                                       Mozilla JS exte
        inpField.value = ""
112
113
         [timeTag, wpmTag, mistakeTag, cpmTag].forEach(tag => tag.innerText = 0);
                                                                                                   31 'const' is availa
114
        loadParagraph();
                                                                                                      Mozilla JS exte
115
      }
116
                                                                                                   34 'let' is available
117
      // End the game
                                                                                                      Mozilla JS exte
118
      function finishGame() {
119
        clearInterval(timer);
inpField.value = ""; const wpm = Math.max(Math.round(((charIndex - mistakes) / !
                                                                                                   35 'let' is available
120
                                                                                                      Mozilla JS exte
121
122
        typingText.innerHTML =
                                                                                                   36 'let' is available
123
          Mozilla JS exte
124
           Your WPM is:
125
      }
                                                                                                   37 'let' is available
126
127
                                                                                                      Mozilla JS exte
      // Starting/initializing the game
128
      loadParagraph();
                                                                                                   38 'let' is available
129
      inpField.addEventListener("input", handleTyping);
      tryAgainBtn.addEventListener("click", resetGame);
130
                                                                                                      Mozilla JS exte
131
                                                                                                   39 'let' is available
132
      // Keydown sound effect
      const sound = new Audio('assets/audio/keydown.wav');
133
                                                                                                      Mozilla JS exte
134
      let soundEnabled = true;
135
                                                                                                   43 'const' is availa
      // Eventlistener for keydown sound effect
136
                                                                                                      Mozilla JS exte
137
      document.addEventListener('keydown', () => {
138
        if (soundEnabled) {
                                                                                                   44 'spread operator
139
           sound.currentTime = 0;
                                                                                                       'esversion: 6').
140
           sound.play();
141
                                                                                                   45 Homplate literal
142
      });
```

https://jshint.com 2/3

```
143
144    const toggleButton = document.getElementById('toggle-sound');
145    toggleButton.addEventListener('click', () => {
146         soundEnabled = !soundEnabled;
147         toggleButton.textContent = soundEnabled ? 'Keyboard Sound ON' : 'Keyboard Sound 148 });
```



version 2.13.6 There are 13 function (https://github.com/jshint/js Function with the larç

About (While the median is 0

Documentation (ABCS) has median is 3.

Install (/install)
The most complex fu
Contribute (/contribute) the

Blog (/blog)

35 warnings

- 2 'const' is availa Mozilla JS exte
- 25 'const' is availa Mozilla JS exte
- 26 'const' is availa Mozilla JS exte
- 27 'const' is availa Mozilla JS exte
- 28 'const' is availa Mozilla JS exte
- 29 'const' is availa Mozilla JS exte
- 30 'const' is availa Mozilla JS exte
- 31 'const' is availa Mozilla JS exte
- 34 'let' is available Mozilla JS exte
- 35 'let' is available Mozilla JS exte
- 36 'let' is available Mozilla JS exte
- 37 'let' is available Mozilla JS exte
- 38 'let' is available Mozilla JS exte
- 39 'let' is available Mozilla JS exte
- 43 'const' is availa Mozilla JS exte
- 44 'spread operato' 'esversion: 6').
- 45 Homplate literal

https://jshint.com 3/3