```
// Auto-Generated Random Paragraphs for typing
const paragraphs = [
    "Typing fast with all ten fingers is a skill every programmer should master. I

    "Programming is not just about writing code but doing it effectively. When you

    "Becoming a fast typist begins with using the correct typing posture. Sit up s

    "Learning programming and touch typing go hand in hand. As you practice coding

    "The journey to mastering typing involves consistency and patience. Begin by p

    "Did you know that the world's fastest typists often use the same strategies a

    "Touch typing is not only a skill for speed but also for multitasking. Program

    "Fast typing doesn't mean rushing through code. Instead, it’s about maintainin

    "Typing with ten fingers boosts productivity and reduces stress during coding r

    "Programming and typing are like two sides of the same coin. When you improve c
  ];

// Selecting DOM elements
const typingText = document.querySelector(".typing-text p");
const inpField = document.querySelector(".wrapper .input-field");
const tryAgainBtn = document.querySelector(".content button");
const timeTag = document.querySelector(".time span b");
const mistakeTag = document.querySelector(".mistake span");
const wpmTag = document.querySelector(".wpm span");
const cpmTag = document.querySelector(".cpm span");

// Declaring variables for game logic
let timer;
let maxTime = 60;
let timeLeft = maxTime;
let charIndex = 0;
let mistakes = 0;
let isTyping = false;

// Load a random paragraph and display it
function loadParagraph() {
  const randomIndex = Math.floor(Math.random() * paragraphs.length);
  typingText.innerHTML = [...paragraphs[randomIndex]]
    .map(char => `<span>${char}</span>`)
    .join("");

  typingText.firstElementChild.classList.add("active");

  // Automatically focus the input field on keydown or click
  const focusInput = () => inpField.focus();
  document.addEventListener("keydown", focusInput);
  typingText.addEventListener("click", focusInput);
}

// Typing Logic Function
function handleTyping() {
  const characters = typingText.querySelectorAll("span");
  const typedChar = inpField.value[charIndex];

  if (charIndex >= characters.length || timeLeft <= 0) return finishGame();

  if (!isTyping) startTimer();

  //Allowing user to backspace
  if (typedChar == null) {
      if (charIndex > 0) {
          charIndex--; // Move back to the previous character
          if (characters[charIndex].classList.contains("incorrect")) mistakes--;
          characters[charIndex].classList.remove("correct", "incorrect", "active")
```

CONFIGURE

version 2.13.6
(https://github.com/jshint/js

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

Metrics

There are 13 functior
Function with the larg
while the median is C
Largest function has
median is 3.
The most complex fu
value of 10 while the

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 'template literal

```
      }

  } else { // Check Character if correct or incorrect
      const isCorrect = characters[charIndex].innerText === typedChar;
      characters[charIndex].classList.add(isCorrect ? "correct" : "incorrect");
      if (!isCorrect) mistakes++;
      charIndex++;
  }

//Update activated character
  characters.forEach(span => span.classList.remove("active"));
  if (charIndex < characters.length) characters[charIndex].classList.add("active")

  updateStats();
}
 // Timer function
function startTimer() {
  isTyping = true;
  timer = setInterval(() => {
      if (--timeLeft >= 0) {
          timeTag.innerText = timeLeft;
          updateStats(); // Update WPM during countdown
      } else {
          clearInterval(timer); // Stopping the timer
          finishGame();
      }
  }, 1000);
}

// Update WPM, CPM, and mistakes
function updateStats() {
  const wpm = Math.max(Math.round(((charIndex - mistakes) / 5) / ((maxTime - timeL
  wpmTag.innerText = wpm;
  cpmTag.innerText = charIndex - mistakes;
  mistakeTag.innerText = mistakes;
}

// Reset the game
function resetGame() {
  clearInterval(timer);
  [timeLeft, charIndex, mistakes, isTyping] = [maxTime, 0, 0, false];
  inpField.value = "";
  [timeTag, wpmTag, mistakeTag, cpmTag].forEach(tag => tag.innerText = 0);
  loadParagraph();
}

// End the game
function finishGame() {
  clearInterval(timer);
  inpField.value = "";  const wpm = Math.max(Math.round(((charIndex - mistakes) /

  typingText.innerHTML = `
    <p style="text-align: center; font-size: 1.5rem; font-weight: bold; color: #ff
    <p style="text-align: center; font-size: 1.5rem; color: #00FF00;">Your WPM is:
}

// Starting/initializing the game
loadParagraph();
inpField.addEventListener("input", handleTyping);
tryAgainBtn.addEventListener("click", resetGame);

// Keydown sound effect
const sound = new Audio('assets/audio/keydown.wav');
let soundEnabled = true;

// Eventlistener for keydown sound effect
document.addEventListener('keydown', () => {
  if (soundEnabled) {
    sound.currentTime = 0;
    sound.play();
  }
});
```

version 2.13.6
(https://github.com/jshint/js

About (/about)
Documentation (/docs)
Install (/install)
Contribute (/contribute)
Blog (/blog)

## Metrics

There are 13 functio
Function with the larg
while the median is 0
Largest function has
median is 3.
The most complex fu
value of 10 while the

## 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 'template literal

```

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

JSHint

version 2.13.6
(https://github.com/jshint/js

About (/about)

Documentation (/docs)

Install (/install)

Contribute (/contribute)

Blog (/blog)

Metrics

There are 13 functior
Function with the larg
while the median is C
Largest function has
median is 3.
The most complex fu
value of 10 while the

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 'template literal