

## Rewrite this code by replacing the arrow functions with regular ones: // Step 7: Remove a box on double-click

```
document.addEventListener("dblclick", (event) => {
  if (event.target.classList.contains("box")) {
    event.target.remove();
  }
});

// Step 8: Display a box's page coordinates on hover
document.addEventListener("mouseover", (event) => {
  if (event.target.classList.contains("box")) {
    const rect = event.target.getBoundingClientRect();
    const coordinates = `(${Math.round(rect.left)}, ${Math.round(rect.top)})`;
    event.target.textContent = coordinates;
  }
});

// Step 9: Display a box's ID back when the mouse leaves
document.addEventListener("mouseout", (event) => {
  if (event.target.classList.contains("box")) {
    event.target.textContent = event.target.dataset.id;
  }
});

// Step 10: Create a new box when the "N" key is pressed
document.addEventListener("keydown", (event) => {
  if (
    (event.key === "N" || event.key === "n") &&
    document.activeElement !== colorInput
  ) {
    addNewBox();
  }
});
```

});

Here's the code rewritten with regular functions instead of arrow functions:

```
// Step 7: Remove a box on double-click
document.addEventListener("dblclick", function(event) {
  if (event.target.classList.contains("box")) {
    event.target.remove();
  }
});
```

```

});

// Step 8: Display a box's page coordinates on hover
document.addEventListener("mouseover", function(event) {
  if (event.target.classList.contains("box")) {
    const rect = event.target.getBoundingClientRect();
    const coordinates = `(${Math.round(rect.left)}, ${Math.round(rect.top)})`;
    event.target.textContent = coordinates;
  }
});

// Step 9: Display a box's ID back when the mouse leaves
document.addEventListener("mouseout", function(event) {
  if (event.target.classList.contains("box")) {
    event.target.textContent = event.target.dataset.id;
  }
});

// Step 10: Create a new box when the "N" key is pressed
document.addEventListener("keydown", function(event) {
  if (
    (event.key === "N" || event.key === "n") &&
    document.activeElement !== colorInput
  ) {
    addNewBox();
  }
});

```

In this rewritten version, each arrow function has been replaced with a regular function using the `function` keyword<sup>[1] [2]</sup>. The functionality remains the same, but the syntax has been changed to use traditional function expressions instead of arrow functions<sup>[3] [4]</sup>.

\*\*

1. <https://forum.freecodecamp.org/t/arrow-function-how-to-rewrite-to-regular-function/417198>
2. [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow\\_functions](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Functions/Arrow_functions)
3. <https://www.freecodecamp.org/news/javascript-arrow-functions-in-depth/>
4. <https://stackoverflow.com/questions/37668185/converting-ecmascript-6s-arrow-function-to-a-regular-function>