How to write clean code JavaScript indentations and spacing

```
function greet(name) {
    if (name) {
        console.log("Hello, " + name + "!");
    } else {
        console.log("Hello, world!");
    }
}
```


Indentation and whitespace are crucial aspects of coding in JavaScript, as they help improve code readability and maintainability. Properly formatted code is easier to understand and debug. In JavaScript, indentation and whitespace are primarily used to structure your code, making it more organized and visually appealing.

Are you a JavaScript enthusiast looking to level up your coding skills? One key aspect often overlooked but crucial for writing clean and maintainable code is proper indentation and whitespace. Here's why it matters:

- Readability: Indentation helps structure your code, making it easier to understand. Well-placed spaces and line breaks create a visual hierarchy, highlighting code blocks, loops, and function definitions.
- Maintainability: Clean, properly formatted code is a breeze to maintain. It reduces the likelihood of bugs and makes debugging faster and more efficient.
- Collaboration: Consistent code formatting is essential when working on a team. It ensures everyone can easily read, understand, and contribute to the codebase.
- Professionalism: Employing best practices in indentation and whitespace showcases your coding discipline and professionalism as a developer.

Remember, while JavaScript itself doesn't enforce specific formatting rules, adhering to established coding style guidelines is key. Whether it's spaces or tabs, proper spacing around operators, or well-placed line breaks, these habits elevate your coding game.

So, JavaScript learners, take the time to practice these foundational skills. They might seem small, but they make a big difference in the quality of your code and your journey to becoming a proficient developer. 👨‍💻👨‍💻

Indentation:

Indentation refers to the spaces or tabs added at the beginning of each line of code to create a visual hierarchy within your code. It helps you identify code blocks, loops, and function definitions. Typically, developers use either spaces or tabs for indentation, but it's important to choose one and stick with it throughout your codebase for consistency. The most common convention is to use four spaces for each level of indentation.

Here's an example of JavaScript code with proper indentation:

```
function greet(name) {
    if (name) {
        console.log("Hello, " + name + "!");
    } else {
        console.log("Hello, world!");
    }
}
```

In this example, the if statement and the code blocks within the if and else branches are indented to show their hierarchical relationship.

Whitespace:

Whitespace in JavaScript refers to spaces, tabs, line breaks, and other non-printable characters. Whitespace is used to separate tokens (such as variables, operators, and keywords) to make the code more readable. While excessive whitespace is unnecessary, proper spacing can significantly enhance code clarity.

Here are some common uses of whitespace in JavaScript:

Spacing around operators: Adding spaces around operators makes expressions more readable:

```
let sum = 3 + 4; // Good
let sum=3+4;    // Avoid
```

Spacing after commas:

Use spaces after commas in function arguments, array elements, and object properties:

```
let colors = ["red", "green", "blue"]; // Good
let colors = ["red","green","blue"];   // Avoid
```

Line breaks:

Use line breaks to separate blocks of code, especially for long or complex statements:

```
function calculateTotal(price, quantity) {
    return price * quantity;
}
```

Spacing in function calls:

Use spaces between function names and their parentheses for better readability:

```
console.log("Hello, world!"); // Good
console.log ("Hello, world!"); // Avoid
```

Indentation with whitespace:

As mentioned earlier, use spaces or tabs consistently for indentation to align code blocks properly.

Empty lines:

Use empty lines to separate logical sections of your code to improve readability.

```
function operationA() {
    // code for operation A
}

function operationB() {
    // code for operation B
}
```

Consistency in your use of whitespace is essential. Most code editors and IDEs have built-in features or plugins that can help automatically format your code according to established conventions.

Remember that while JavaScript itself doesn't enforce specific formatting rules, adhering to common coding style guidelines, such as those provided by organizations like the JavaScript Standard Style or Airbnb's JavaScript Style Guide, can help ensure your code is clean and maintainable, making it easier to collaborate with other developers.

Practice exercise

1. **Indentation Practice:**

   Write a function that takes a number as input and prints all the even numbers from 1 to that number. Make sure to use proper indentation to structure your code.

2. **Whitespace in Expressions:**

   Create a function that calculates the area of a rectangle. The function should take two arguments, length and width, and return the area. Use proper spacing around operators.

3. **Line Breaks:**

   Write a program that prints the first ten Fibonacci numbers on separate lines. Use line breaks to separate each number in the output.

4. **Spacing in Function Calls:**

   Define a function called calculateCircleArea that takes the radius as a parameter and returns the area of a circle. Call this function with different radii, and ensure you use proper spacing around the function name and parentheses.

5. **Indentation and Conditional Statements:**

   Write a function that determines whether a given number is positive, negative, or zero. Use proper indentation to structure the if-else statements.

6. **Whitespace in Arrays:**

   Create an array of your favorite fruits. Add a few more fruits to the array, and ensure there are spaces after commas between the fruit names.

7. **Spacing in Object Properties:**

   Define an object called person with properties firstName and lastName. Use proper spacing between property names and values.

8. **Indentation and Loops:**

   Write a program that prints the multiplication table for a given number (e.g., 5) from 1 to 10. Use proper indentation within the loop.

9. **Whitespace in Function Definitions:**

   Define a function called greetUser that takes a name as an argument and prints a greeting message. Ensure proper spacing in the function definition.

10. **Empty Lines:**

    Create a simple JavaScript program that simulates a shopping cart. Use empty lines to separate different parts of the program, such as defining variables, adding items to the cart, and calculating the total cost.

Feel free to modify these exercises or create your own to practice indentation and whitespace in JavaScript. Proper formatting is essential for writing clean and maintainable code, and these exercises can help you develop good coding habits.