Clean code principles

1. Magic Numbers

A *magic number* means we are assigning a number with no clear meaning. Sometimes we use a value for a specific purpose, and we don't assign the value in a meaningful variable. The problem is that when someone works with your code, then the person doesn't know the meaning of that direct value.

Magic numbers

```
{//Bad
practice
      for(let i = 0; i < 50; i++){
        //do something
      }
      //Good practice
      let NUMBER_OF_STUDENTS= 50
      for(let i = 0; i < NUMBER_OF_STUDENTS; i++){
        //do something
      }
```

2-Use Pronounceable Names

If you can't pronounce a name, you can't discuss it without sounding silly.

Bad:

```
const yyyymmdstr = moment().format("YYYY/MM/DD");
```

Good:

```
const currentDate = moment().format("YYYY/MM/DD");
```

3. Comments

This one is a kind of personal attack on someone. Comments help people to understand later, and they help other programmers to work on the same project. Comments in code mean maybe your code is not self-explanatory. Here is a famous quote about writing comments by Jeff Atwood.Comments should be good, but your code needs to be self-explanatory.

4. Avoid Large Functions

When a function or a class is much larger, then it is suggested to separate it into multiples. This will make our code easier, clean, easy to understand, and also reusable……Suppose we need to add and subtract two numbers. We can do it with a single function. But the good practice is to divide them into two. When there are individual functions, then this will be reusable in the whole application.

```
// good practice
// add
const add = (a,b) => {
    return a+b
}
// sub
const sub = (a,b) => {
    return a-b
}
```

5. Code Repetition

Repeated code means a code block that is repeated in your code more than once. This means your code portion needs to be extracted into a function.

6. Meaningful Names

A meaningful name is one of the most important conventions. Always use a meaningful name for variables, functions, and others. Choose a name that expresses the meaning of your purpose.

7. Avoid One-Letter Variable Names

A one-letter variable is a very, very bad thing to use. Don't use this for a variable name.But in a loop, we use some variables with a letter, which is OK to use.

8-Deep Nesting

Sometimes we use nested loops that are difficult to understand. The way to handle that is to extract all loops into separate functions instead.

9- Favor Descriptive Over Concise

Try to use detail for any naming. Suppose we need a function that will find a user with their phone. Here we can use meaningful names, but there is a huge possibility of mistakes if there are other, similar functions.

We must use a detailed, meaningful name that expresses the meaning in a nutshell.

```
//We want a function for search user against phone no//Bad practice
const searchUser = (phone) => {
//Do something
}//Good practice
const searchUserByPhoneNo = (phone) => {
//Do something
}
```

10-Functions

They should be small.
They should do only one thing and they should do it well
Use descriptive name

11-Avoid Noise Words

Noise words are the words that do not offer any additional information about the variable. They are redundant and should be removed.

Some popular noise words are:

- The (prefix)
- Info
- Data
- Variable
- Object
- Manager

If your class is named UserInfo, you can just remove the Info and make it User. Using BookData instead of Book as class name is just a no-brainer, as a class stores Data anyways.

12-Write Readable Code For People

A lot of people especially beginners make mistake while writing a code they write everything in a single line and don’t give proper whitespace, indentation or line breaks in their code. It makes their code messy and difficult to maintain. There’s always a chance that another human will get to your code and they will have to work with it. It wastes other developers’ time when they try to read and understand the messy code. So always pay attention to the formatting of your code. You will also save your time and energy when you will get back to your own code after a couple of days to make some changes. So make sure that your code should have a proper indentation, space and line breaks to make it readable for other people. The coding style and formatting affect the maintainability of your code.

13-Use Nouns for Class Name

Classes don't take things; they are the things. Class is mainly a blueprint for something. Don't use the verb in the class name.

Also, a class should contain Pascal case. Camel case is used for objects, so this won't be very clear if you use camel case for class.

```
//bad practice
class MakeCar = {
   //...
}//Good practice
class Car = {
   //...
}
```