Coding Conventions:

Naming:

- Common identifier:
  - common variables names are written in **lowerCamelCase**
  - eg. rightCount,customerId,nextPage.
- Package names:
  - Package names are all **lowerCamelCase**. For example, `my.exampleCode.deepSpace`
- Class name:
  - Class, interface, record, and typedef names are written in **UpperCamelCase**
  - `eg.Requset,MyClass,Runable,NewMode`
- Method names:
  - Method names are written in **lowerCamelCase**
  - eg.sendMessage,getFoo,isFoo,hasFoo,setValue(value)
- Constant names:
  - Constant names use **CONSTANT_CASE**: all uppercase letters, with words separated by underscores.
  - eg.CONSTANT_NAME.

Code Indentation:

- Always use **2 spaces** for indentation of code blocks:

  Eg.

  Example

```
function toCelsius(fahrenheit) {
  return (5 / 9) * (fahrenheit - 32);
}
```

Line Lenght And Line Breaking:

- Lines should usually be no longer than **80 characters**, and should not exceed 100 (counting tabs as 4 spaces). *This is a "soft" rule, but long lines generally indicate unreadable or disorganized code.*
- the best place to break it is after an operator or a comma.
- **Example**

```
callAFunction(document, element, window, "some string value", true, 123,
        navigator);
```

Statement Rules:

- Always end a simple statement with a semicolon.
- Put the opening bracket at the end of the first line.
- Use one space before the opening bracket.
- Put the closing bracket on a new line, without leading spaces.
- Do not end a complex statement with a semicolon.
- Do not omit bracket even its a one-line statement

Example

```
//bad! may cause error when expand
if (condition)
doSomething();

// good example
if (time < 20) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
```

Spaces Around Operators:

- Always put spaces around operators (= + - * /), and after commas:
- **Example**

```
var x = y + z;
var values = ["Volvo", "Saab", "Fiat"];
```

Objects:

- Object declarations can be made on a single line if they are short.
- When an object declaration is too long to fit on one line, there must be one property per line.
- When an object or array is too long to fit on one line, each member must be placed on its own line and each line ended by a comma.
- **example**

```
// preferred
var obj = {
    ready: 9,
    when: 4,
    'you are': 15,
};
var arr = [
    9,
    4,
    15,
];

// Acceptable for small objects and arrays
var obj = { ready: 9, when: 4, 'you are': 15 };
var arr = [ 9, 4, 15 ];

// Bad
var obj = { ready: 9,
    when: 4, 'you are': 15 };
var arr = [ 9,
    4, 15 ];
```

Comments:

- Comments come before the code to which they refer, and should always be preceded by a blank line.
- Capitalize the first letter of the comment.
- Each class should contain a comment indicating its purpose with the author, creating time.
- Every method must contain a comment at the beginning explaining its behavior.

Example

```
/**
* This class is for doing something.
* @ Author
* Created on 20 March, 2020
**/

someStatement();

// Explanation of something complex on the next line
$( 'p' ).doSomething();

// This is a comment that is long enough to warrant being stretched
// over the span of multiple lines.
```