# JavaScript Doc Guide

JSDoc is an API documentation generator for JavaScript, similar to Javadoc or phpDocumentor. You add documentation comments directly to your source code, right alongside the code itself. The JSDoc tool will scan your source code and generate an HTML documentation website for you.

Add JsDoc comments to your code if this is necessary(in JS is really necessary!!!).

# @class

Use this tag for define class and add comment

/\*\*  
 \* Creates a new Person.  
 \* @class  
 \*/  
class Person {  
}  
  
const p = new Person();

# @param

This tag provides the name, type, and description of a function/method parameter.

/\*\*  
 \* @param {string} somebody Somebody's name.  
 \*/  
function sayHello(somebody) {  
    console.log('Hello ' + somebody);  
}

# @private @public @protected

This tag marks class/property/method as a private/public/protected.

/\*\*  
 \* @public  
 \*/  
class Person {

construtor() {

/\*\*  
  \* Age of the person  
  \* @private  
  \*/

this.\_age = 12;

/\*\*  
  \* Name of the person  
  \* @protected  
  \*/

this.\_name ="Jack";

}  
}

# @readonly

This tag indicates that the value can't be changed.

/\*\*  
 \* A constant.  
 \* @readonly  
 \*/  
const foo = 1;

class Person {

constructor() {

this.\_age = 12;

}

/\*\*  
  \* @readonly  
  \*/

get age() {

return this.\_age

}  
}

# @returns

The @returns tag show what parameters function returns.

/\*\*  
 \* Returns the sum of a and b  
 \* @param {number} a  
 \* @param {number} b  
 \* @returns {number}  
 \*/  
function sum(a, b) {  
    return a + b;  
}

# @type

The @type tag allows you to provide a type expression of the variable/property(getter and setter).

/\*\* @type {string|Array} \*/  
var foo;  
/\*\* @type {number} \*/  
var bar = 1;

class Person {

/\*\* @type {number} \*/

get age() {

return this.\_age;

}  
}