Free and open source tools for doc generation:

the JSDoc Toolkit (<http://code.google.com/p/jsdoc-toolkit/>)

and YUIDoc (<http://yuilibrary.com/projects/yuidoc>).

Process of generating API documentation

• Writing specially formatted code blocks

• Running a tool to parse the code and the comments

• Publishing the results of the tool, which are most often HTML pages

[复制代码](javascript:void(0);)

/\*\*

\* Reverse a string

\*

\* @param {String} input String to reverse

\* @return {String} The reversed string

\*/

var reverse = function (input) {

// ...

return output;

};

[复制代码](javascript:void(0);)

 The contents of app.js starts like this:

[复制代码](javascript:void(0);)

/\*\*

\* My JavaScript application

\*

\* @module myapp

\*/

[复制代码](javascript:void(0);)

Then you define a blank object to use as a namespace:

var MYAPP = {};

And then you define an object math\_stuff that has two methods: sum() and multi():

[复制代码](javascript:void(0);)

/\*\*

\* A math utility

\* @namespace MYAPP

\* @class math\_stuff

\*/

MYAPP.math\_stuff = {

/\*\*

\* Sums two numbers

\*

\* @method sum

\* @param {Number} a First number

\* @param {Number} b The second number

\* @return {Number} The sum of the two inputs

\*/

sum: function (a, b) {

return a + b;

},

/\*\*

\* Multiplies two numbers

\*

\* @method multi

\* @param {Number} a First number

\* @param {Number} b The second number

\* @return {Number} The two inputs multiplied

\*/

multi: function (a, b) {

return a \* b;

}

};

[复制代码](javascript:void(0);)

@namespace

The global reference that contains your object.

@class

A misnomer (no classes in JavaScript) that is used to mean either an object or a constructor function.

@method

Defines a method in an object and specifies the method name.

@param

Lists the arguments that a function takes. The types of the parameters are in curly braces, followed by the parameter name and its description.

@return

Like @param, only it describes the value returned by the method and has no name.

• @constructor hints that this “class” is actually a constructor function

• @property and @type describe properties of an object

[复制代码](javascript:void(0);)

/\*\*

\* Constructs Person objects

\* @class Person

\* @constructor

\* @namespace MYAPP

\* @param {String} first First name

\* @param {String} last Last name

\*/

MYAPP.Person = function (first, last) {

/\*\*

\* Name of the person

\* @property first\_name

\* @type String

\*/

this.first\_name = first;

/\*\*

\* Last (family) name of the person

\* @property last\_name

\* @type String

\*/

this.last\_name = last;

};

/\*\*

\* Returns the name of the person object

\*

\* @method getName

\* @return {String} The name of the person

\*/

MYAPP.Person.prototype.getName = function () {

return this.first\_name + ' ' + this.last\_name;

};

[复制代码](javascript:void(0);)

YUIDoc Example

