

# JSDoc3 Cheat Sheet

## Normal Tags

### @abstract | @virtual

### @alias

```
@alias <name>
```

### @author

```
@author <name> [<emailAddress>]
```

### @borrows

This object uses something from another object.

```
@borrows <name> as <name>
```

### @callback

```
@callback <name>
```

```
/**
 * Send a request.
 * @param {requestCallback} cb
 * - Callback handling responses.
 */
MyClass.send = function(cb)
{ //code };

/**
 * This callback is displayed
 * as a global member.
 * @callback requestCallback
 * @param {Number} responseCode
 */
```

### @class | @constructor

```
@class [{{type}} <name>]
```

### @const | @constant

```
@constant [{{type}} <name>]
```

### @constructs

This function member will be the constructor for the previous class.

```
@constructs [<name>]
```

### @copyright

```
@copyright <text>
```

### @default | @defaultvalue

```
@default [<value>]
```

### @deprecated

```
@deprecated [<text>]
```

### @enum

```
@enum [{{type}}]
```

### @event

```
@event
<className>#[event:]<name>
```

```
/**
 * Throw a snowball.
 * @fires Hurl#snowball
 */
Hurl.prototype.snowball =
function() { //code };

/**
 * Snowball event.
 * @event Hurl#snowball
 * @type {Object}
 * @property {Boolean} isPacked
 * - Indicates packed snowball.
 */
```

### @example

```
/**
 * Solves equations.
 * @example <caption>Example
 * usage of method1.</caption>
 * // returns 2
 * globalNS.method1(5, 10);
 * @returns {Number} value of x
 */
```

### @exports

Identify the member that is exported.

```
@exports <name>
```

### @extends | @augments

```
@extends <name>
```

### @external | @host

Document an external class / namespace / module.

```
@external <name>
```

```
/**
 * Provided by the browser.
 * @external XMLHttpRequest
 */

/**
 * Extends the XMLHttpRequest.
 * @class EncryptedRequest
 * @extends
 * external:XMLHttpRequest
 */
```

### @fires | @emits

```
@fires
<className>#[event:]<name>
```

### @function | @method

```
@function [<name>]
```

### @global

### @ignore

Remove this from the final output.

### @inner

### @instance

### @lends

Document properties on an object literal as if they belonged to a symbol with a given name.

```
@lends <name>
```

### @license

```
@license <text>
```

### @member | @var

```
@member [{{type}}] [<name>]
```

### @memberof

```
@memberof <name>
```

```
/** @memberof Class.static */
/** @memberof Class#instance */
/** @memberof Class~inner */
```

### @mixes

This object mixes in all the members from another object.

```
@mixes <name>
```

### @mixin

```
@mixin [<name>]
```

### @module

```
@module [{{<type>}}] <name>]
```

### @name

```
@name <name>
```

### @namespace

```
@namespace [{{<type>}}] <name>]
```

### @param | @argument

```
@param {type} <name>
<description>
@param {type}
[<name>=<defaultValue>]
<description>
```

## @private

## @property | @prop

```
@property {type} <name>
<description>
```

## @protected

## @public

## @readonly

## @requires

```
@requires <name>
```

## @return | @returns

```
@returns {type} <description>
```

## @see

```
@see <name>
@see <text>
```

## Inline tags

### {@link}

Link to a JSDoc namepath or an external URL. By default, links are rendered in plain text. If you prefer links to be rendered in a monospace font, you can use the configuration options `template.monospaceLinks` and `templates.cleverLinks` to change this behavior.

### {@linkcode}

Insert a link that should be rendered in a monospace font.

### {@linkplain}

Insert a link that should not be rendered in a monospace font.

### {@tutorial}

Link to a tutorial.

```
{@link someNamepathOrURL}
[link text here]{@link
someNamepathOrURL}
{@link someNamepathOrURL|link
text here}
{@link someNamepathOrURL Link
text here (after the first
space)}
```

```
/**
 * A module. Refer to it using
 {@link module:foo/bar}.
 * @module foo/bar
 */
```

## @since

```
@since <version>
```

## @static

## @summary

```
@summary <text>
```

## @this

What does 'this' refer to here?

```
@this <name>
```

## @throws | @exception

```
@throws <description>
@throws {type} [<description>]
```

## @todo

```
@todo <text>
```

## @tutorial

Insert a link to an included tutorial file.

```
@tutorial <tutorialID>
```

```
/**
 * The built-in string object. Refer to it with {@link
external:String}.
 * @external String
 */
/**
 * An event. Refer to with {@link module:foo/bar.event:MyEvent}.
 * @event module:foo/bar.event:MyEvent
 */
```

## @type

```
@type {type}
```

## @typedef

Document a custom type.

```
@typedef [{type}] <name>
```

```
/**
 * @typedef PropHash
 * @type {Object}
 * @property {String} id
 * @property {String} name
 */
/** @type {PropHash} */
var props;
```

## @variation

Distinguish different objects with the same name.

```
@variation <variationNumber>
```

## @version

```
@version <version>
```

## Types

Symbol name (name expression)	{Boolean} {myNamespace.MyClass}
Multiple types (type union)	{(Number Boolean)}
Arrays and objects (type applications and record types)	{Array.<MyClass>} or {MyClass[]} {Object.<String, Number>} {{a: Number, b: String, c}} myObj or {Object} myObj {Number} myObj.a {String} myObj.b {*} myObj.c
Nullable type	{?Number}
Non-nullable type	{!Number}
Variable number of that type	@param {...Number} num
Optional parameter	@param {Number} [foo] or @param {Number=} foo @param {Number} [foo=1]