

You deserve the cloud database that's built to adapt. Try MongoDB Atlas.

ads via Carbon

Jsdoc cheatsheet

Functions

```
/**
 * This is a function.
 *
 * @param {string} n - A string param
 * @return {string} A good string
 *
 * @example
 *
 *     foo('hello')
 */

function foo(n) {
  return n
}
```

See: <https://jsdoc.app/index.html>

Variables

```
/**
 * @type {number}
 */
var F00 = 1

/**
 * @const {number}
 */
const F00 = 1
```

Types

@param {string=} n	Optional
@param {string} [n]	Optional
@param {(string number)} n	Multiple types
@param {*} n	Any type
@param {...string} n	Repeatable arguments
@param {string} [n="hi"]	Optional with default
@param {string[]} n	Array of strings
@return {Promise<string[]>} n	Promise fulfilled by array of strings
See: https://jsdoc.app/tags-type.html	

Typedef

```
/**
 * A song
 * @typedef {Object} Song
 * @property {string} title - The title
 * @property {string} artist - The artist
 * @property {number} year - The year
 */

/**
 * Plays a song
 * @param {Song} song - The {@link Song} to be played
```

Typedef Shorthand

```
/**
 * A song
 * @typedef {{title: string, artist: string, year: number}} Song
 */

/**
 * Plays a song
 * @param {Song} song - The {@link Song} to be played
 */

function play(song) {}
```

See: <https://jsdoc.app/tags-typedef.html>

Other keywords

```
/**
 * @throws {FooException}
 * @private
 * @deprecated
 * @see
 *
 * @function
 * @class
 */
```

See the full list: <https://jsdoc.app/index.html#block-tags>

```
*/

function play(song) {}
```

See: <https://jsdoc.app/tags-typedef.html>

Importing types

```
/**
 * @typedef {import('./Foo').default} Bar
 */

/**
 * @param {Bar} x
 */

function test(x) {}
```

This syntax is [TypeScript-specific](#).

Renaming

```
/*
 * @alias Foo.bar
 * @name Foo.bar
 */
```

Prefer `alias` over `name`. See: <https://jsdoc.app/tags-alias.html>

► **7 Comments** for this cheatsheet. [Write yours!](#)