

Type definitions

Boolean	{boolean}	true/false
Integer	{number}	0
Float	{number}	0.0
Object	{Object}	name
SpecifiedObject	{CustomName}	see @typedef
Array	{Array}	name []
Array of Types	{Type[]}	name {string[]}
String	{string}	name 'test'
Optional string	{string}	[name]
Optional string default value	{string}	[name=test]

@statements

@param {type} name	Defines a parameter
@constructor	Defines a constructor
@deprecated	Function is deprecated
@return {type}	Defines return type

Typedef

```
/**
 * @typedef ObjectOne
 * @type {Object}
 * @property {number} id
 * @property {string} name
 */
```

Param example

```
/**
 * @param {string} one
 * @param {number} two
 */
function x( one, two ) {}
```

Array with types example

```
/**
 * @param {string[]} ones
 */
function x( ones ) {}
```

Optional param example

```
/**
 * @param {string} one
 * @param {number} [two]
 */
function x( one, two ) {
  two = two || 0;
}
```

Optional param default value

```
/**
 * @param {string} one
 * @param {number} [two=121]
 */
function x( one, two ) {
  two = two || 121;
}
```

Multiple types

```
/**
 * @param {string|string[]} one
 */
function x( one, two ) {}
```

Detailed object

```
/**
 * @param {ObjectOne} one
 *
 * @typedef ObjectOne
 * @type {Object}
 * @property {number} id
 * @property {string} name
 */
function x( one ) {
  one.id;
  one.name;
}
```