

JavaScript

VS Code JavaScript (ES6) snippets

Visual Studio Marketplace v1.8.0 installs 3477821 rating average: 4.78/5 (23 ratings)

This extension contains code snippets for JavaScript in ES6 syntax for [Vs Code](#) editor (supports both JavaScript and TypeScript).

Note

All the snippets include the final semicolon ; There is a fork of those snippets [here](#) made by @jmsv where semicolons are not included. So feel free to use them according to your needs.

Sponsors



Request and perform code reviews from inside your IDE. Review any code, even if it's a work-in-progress that hasn't been committed yet, and use jump-to-definition, your favorite keybindings, and other IDE tools.

[Try it free](#)

Installation

In order to install an extension you need to launch the Command Palette (Ctrl + Shift + P or Cmd + Shift + P) and type Extensions. There you have either the option to show already installed snippets or install new ones. Search for *JavaScript (ES6) code snippets* and install it.

Supported languages (file extensions)

- JavaScript (.js)
- TypeScript (.ts)
- JavaScript React (.jsx)
- TypeScript React (.tsx)
- Html (.html)
- Vue (.vue)

Snippets

Below is a list of all available snippets and the triggers of each one. The → means the TAB key.

Import and export

Trigger	Content
imp→	imports entire module <code>import fs from 'fs';</code>
imn→	imports entire module without module name <code>import 'animate.css'</code>
imd→	imports only a portion of the module using destructuring <code>import {rename} from 'fs';</code>
ime→	imports everything as alias from the module <code>import * as localAlias from 'fs';</code>
ima→	imports only a portion of the module as alias <code>import { rename as localRename } from 'fs';</code>
rqr→	require package <code>require('');</code>
req→	require package to const <code>const packageName = require('packageName');</code>
mde→	default module.exports <code>module.exports = {};</code>
env→	exports name variable <code>export const nameVariable = localVariable;</code>
enf→	exports name function <code>export const log = (parameter) => { console.log(parameter);};</code>

Trigger	Content
edf→	exports default function export default function fileName (parameter){ console.log(parameter)};
ec1→	exports default class export default class Calculator { };
ece→	exports default class by extending a base one export default class Calculator extends BaseClass { };

Class helpers

Trigger	Content
con→	adds default constructor in the class constructor() {}
met→	creates a method inside a class add() {}
pge→	creates a getter property get propertyName() {return value;}
pse→	creates a setter property set propertyName(value) {}

Various methods

Trigger	Content
fre→	forEach loop in ES6 syntax array.forEach(currentItem => {})
fof→	for ... of loop for(const item of object) {}
fin→	for ... in loop for(const item in object) {}
anfn→	creates an anonymous function (params) => {}
nfn→	creates a named function const add = (params) => {}
dob→	destructuring object syntax const {rename} = fs
dar→	destructuring array syntax const [first, second] = [1,2]
sti→	set interval helper method setInterval(() => {});
sto→	set timeout helper method setTimeout(() => {});
prom→	creates a new Promise return new Promise((resolve, reject) => {});
thenc→	adds then and catch declaration to a promise .then((res) => {}).catch((err) => {});

Console methods

Trigger	Content
cas→	console alert method console.assert(expression, object)
ccl→	console clear console.clear()
cco→	console count console.count(label)
cdb→	console debug console.debug(object)
cdi→	console dir console.dir
cer→	console error console.error(object)
cgr→	console group console.group(label)
cge→	console groupEnd console.groupEnd()
clg→	console log console.log(object)
clo→	console log object with name console.log('object :>> ', object);
ctr→	console trace console.trace(object)
cwa→	console warn console.warn
cin→	console info console.info
clt→	console table console.table
cti→	console time console.time
cte→	console timeEnd console.timeEnd