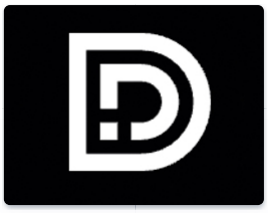
DEVHINTS.IO

Edit

ES2015+ cheatsheet



A quick overview of new JavaScript features in ES2015, ES2016, ES2017, ES2018 and beyond.

Block scoping

Let

```
function fn () {

  if (true) {

  }
}
```

Const

```
const a = 1
```

`let` is the new `var`. Constants work just like `let`, but can't be reassigned. See: Let and const

New methods

New string methods

```
"hello".repeat(3)
"hello".includes("ll")
"hello".startsWith("he")
"hello".padStart(8) // "   hello"
"hello".padEnd(8) // "hello   "
"hello".padEnd(8, '!') // hello!!!
"\u1E9B\u0323".normalize("NFC")
```

See: New methods

Backtick

Interpolat

```
const m
```

Multiline

```
const s
hello
world
`
```

Templat

Binary a

```
let bin
let oct
```

Bin

Classes

```
class C
```

Construct

```
// Same as: Math.pow(2, 8)
```

```
ret
}
```

Calling su

```
expan
}
```

Static me

```
ret
}
}
```

Syntacti

Promises

Making promises

```
  if (ok) { resolve(result) }
  else { reject(error) }
})
```

For asynchronous programming. See: Promises

Using promises

```
promise
```

Promise functions

```
Promise.all(···)
Promise.race(···)
Promise.reject(···)
```

Async-await

```
async function run () {

  return [user, tweets]
}
```

async functions are another way of using functions.

See: async function

Destructuring

Destructuring assignment

Arrays

Objects

```
  title: 'The Silkworm',
  author: 'R. Galbraith'
}
```

Supports for matching arrays and objects. See: Destructuring

Default values

```
const scores = [22, 33]
const [math = 50, sci = 50, arts =
```

```
// Result:
// math === 22, sci === 33, arts =
```

Default values can be assigned while dest

Default values

```
  console.log(`Hi ${name}!`);
}
```

Loops

```
  ...
}
```

The assignment expressions work in loops, too.

ar

Object destructuring

Extract some keys individually and remair

Spread

with Object spread

```
const options = {

  visible: true
}
```

without Object spread

```
const options = Object.assign(
  {}, defaults,
  { visible: true })
```

The Object spread operator lets you build new objects from other objects.

See: Object spread

with Array

```
const u


  'rsta
]
```

without A

```
const u
  .conc
  .conc
```

The spre

See: Spr

Functions

Function arguments

Default arguments

```
  return `Hello ${name}`
}
```

Rest arguments

```
  // y is an Array
  return x * y.length
}
```

Spread

```
// same as fn(1, 2, 3)
```

Default, rest, spread. See: Function arguments

Fat arro

Fat arrow

```
  ...
})
```

With argu

```
  ...
})
```

Implicit re

Objects

Shorthand syntax

```
module.exports = { hello, bye }
// Same as: module.exports = { hello: hello, bye: bye }
```

See: Object literal enhancements

Getters and setters

```
const App = {

    return this.status === 'closed'
  },

    this.status = value ? 'closed' : 'open'
  }
}
```

See: Object literal enhancements

Extract values

```
const fatherJS = { age: 57, name: "Brendan Eich" }


// [57, "Brendan Eich"]

// [["age", 57], ["name", "Brendan Eich"]]
```

Modules

```
import 'helpers'
// aka: require('···')
```

```
export
// aka:
```

```
import Express from 'express'
// aka: const Express = require('···').default || require('···')
```

```
export
// aka:
```

```
import { indent } from 'helpers'
// aka: const indent = require('···').indent
```

```
export
// aka:
```

```
import * as Helpers from 'helpers'
// aka: const Helpers = require('···')
```

```
export i
```

```
import { indentSpaces as indent } from 'helpers'
// aka: const indent = require('···').indentSpaces
```

`import` is the new `require()`. See: Module imports

♯ Generators

Generators

```
function* idMaker () {
  let id = 0
  while (true) { yield id++ }
}
```

```
let gen = idMaker()
gen.next().value  // → 0
gen.next().value  // → 1
gen.next().value  // → 2
```

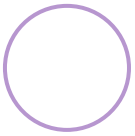
It's complicated. See: Generators

For..of it

```
for (le
  ···
}
```

For itera

Search 358+ cheatsheets



Over 358 curated cheatsheets, by developers for developers.

Devhints home

Other JavaScript cheatsheets

- Vue.js cheatsheet
- JavaScript Date cheatsheet
- fetch() cheatsheet
- JavaScript speech synthesis cheatsheet
- Jsdoc cheatsheet
- npm cheatsheet

Top cheatsheets

- Elixir cheatsheet
- React.js cheatsheet
- Vimdiff cheatsheet
- Vim cheatsheet
- Vim scripting cheatsheet
- Vue.js cheatsheet