`tt()`

Schedule

1. Review assignments (hof, scope, pure)
2. Modules (import / export)
3. Dataset research
4. All together!

Schedule

1. **Review assignments (hof, scope, pure)**
2. Modules (import / export)
3. Dataset research
4. All together!

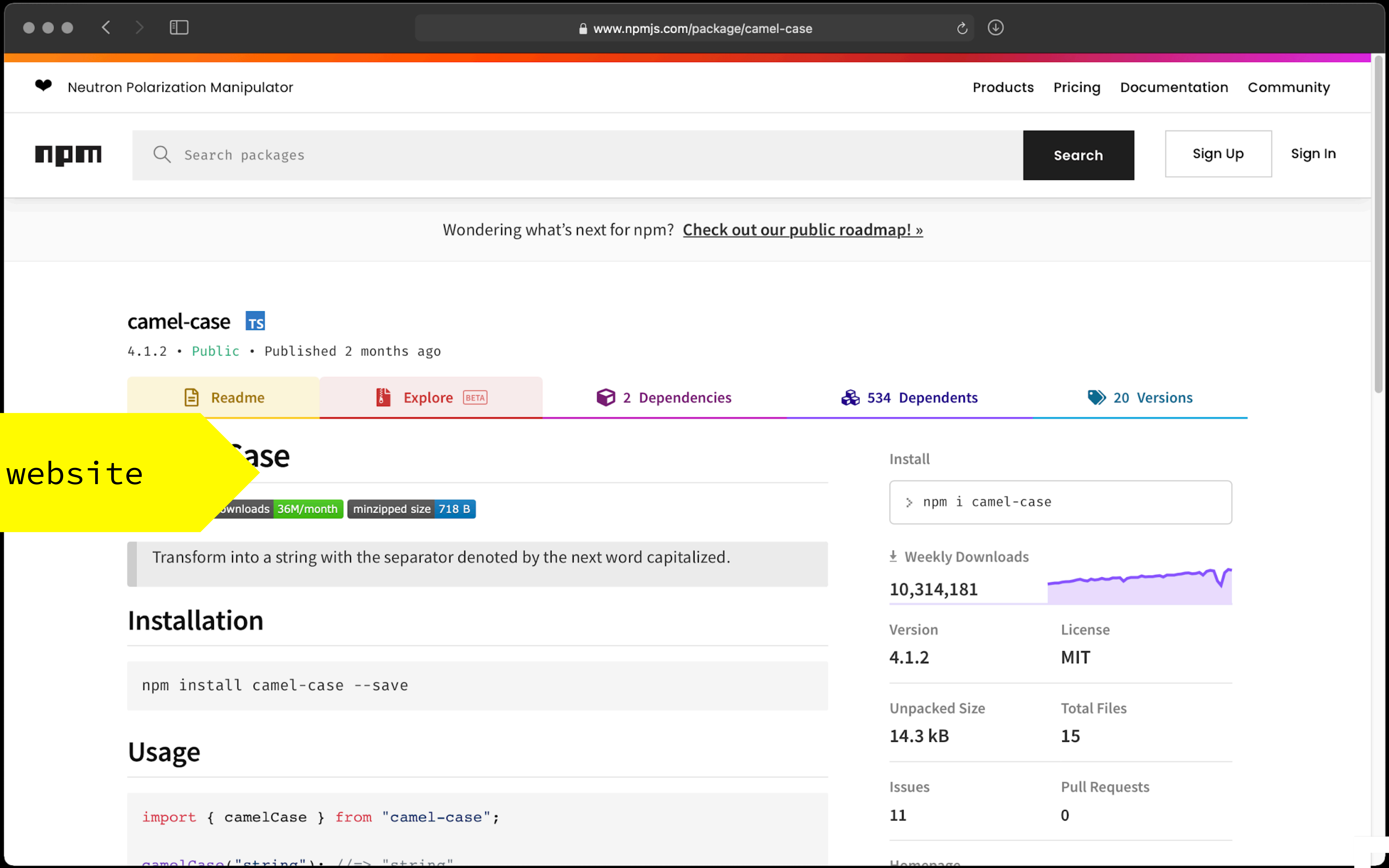
1

Schedule

1. Review assignments (hof, scope, pure)
2. **Modules (import / export)**
3. Dataset research
4. All together!

Package manager

npm is a package manager for the JavaScript programming language. It is the default package manager for […] Node.js. It consists of a command line client, also called npm, and an online database of […] packages, called the npm registry.


www.npmjs.com/package/camel-case

Neutron Polarization Manipulator

Products Pricing Documentation Community

Search packages | Search | Sign Up | Sign In

Wondering what's next for npm? Check out our public roadmap! »

camel-case TS

4.1.2 • Public • Published 2 months ago

Readme | Explore BETA | 2 Dependencies | 534 Dependents | 20 Versions

ase

ownloads 36M/month | minzipped size 718 B

Transform into a string with the separator denoted by the next word capitalized.

Installation

```
npm install camel-case --save
```

Usage

```
import { camelCase } from "camel-case";
```

Install

```
> npm i camel-case
```

Weekly Downloads
10,314,181

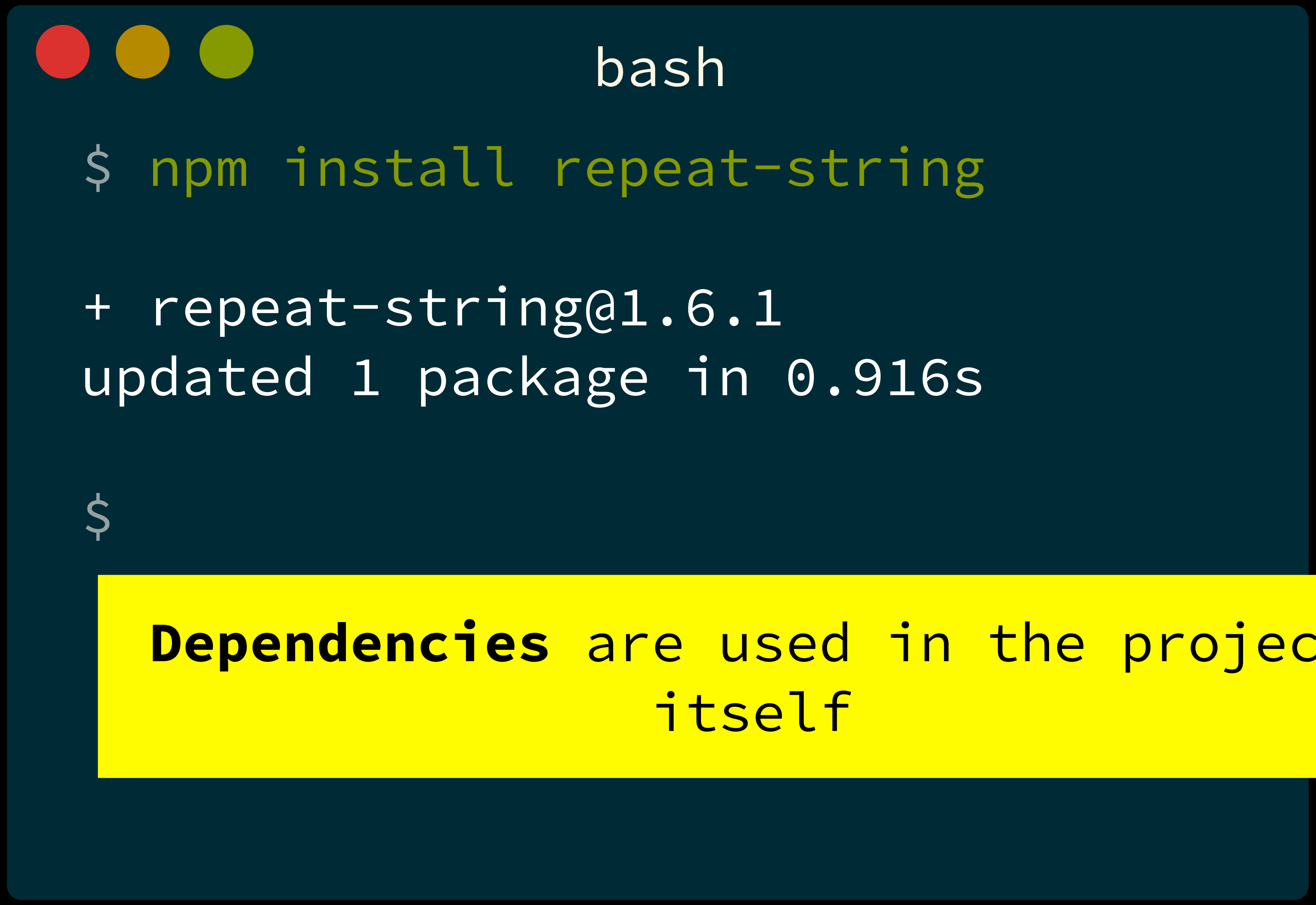
Version	License
4.1.2	MIT

Unpacked Size	Total Files
14.3 kB	15

Issues	Pull Requests
11	0

The **npm** website

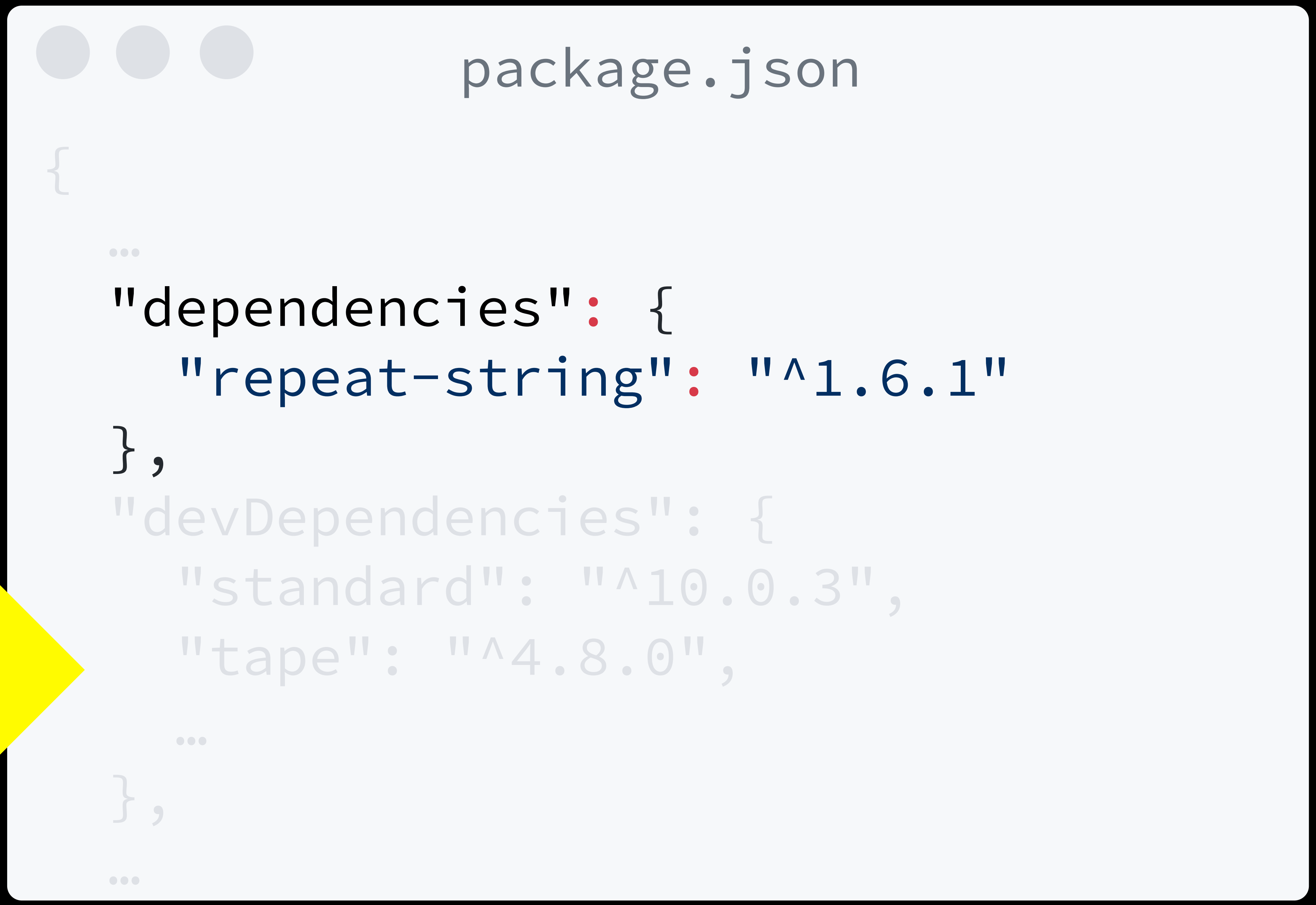
NodeJS



bash

```
$ npm install repeat-string

+ repeat-string@1.6.1
updated 1 package in 0.916s

$
```

Dependencies are used in the project itself



package.json

```
{
  …
  "dependencies": {
    "repeat-string": "^1.6.1"
  },
  "devDependencies": {
    "standard": "^10.0.3",
    "tape": "^4.8.0",
    …
  },
  …
```

NPM

NPM is a database of JavaScript modules, some for back-end, others for front-end.

D3 (the library we’ll be using next week) is hosted here as well for example.

Functions

Functions enable functionality (no way). We use functions inside modules to handle individual tasks. We seek to use functions only for a single task in order to make them easily reusable. For instance, the function getData() can grab multiple sources of data, depending on the provided URL.

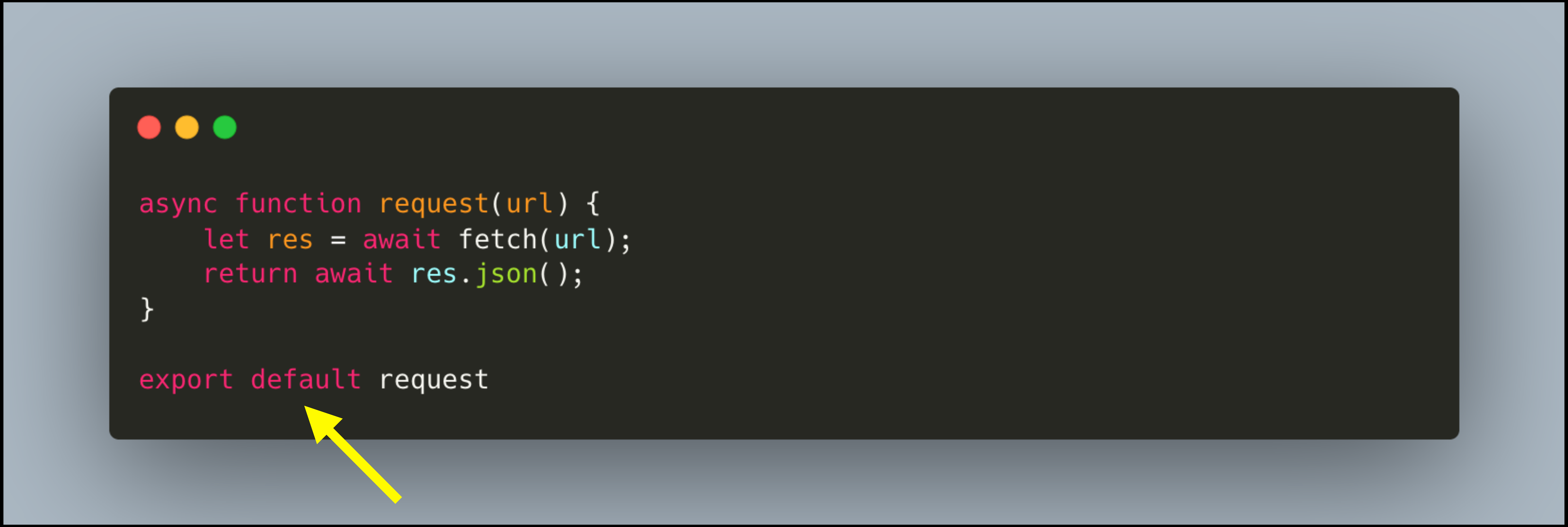
Components

Components also are used for modularization: a way to divide a codebase into separate blocks.

In Front End Development, we use User Interface components to define (preferably) reusable UI elements.

Svelte is component-based. Svelte components combine HTML, CSS and JavaScript

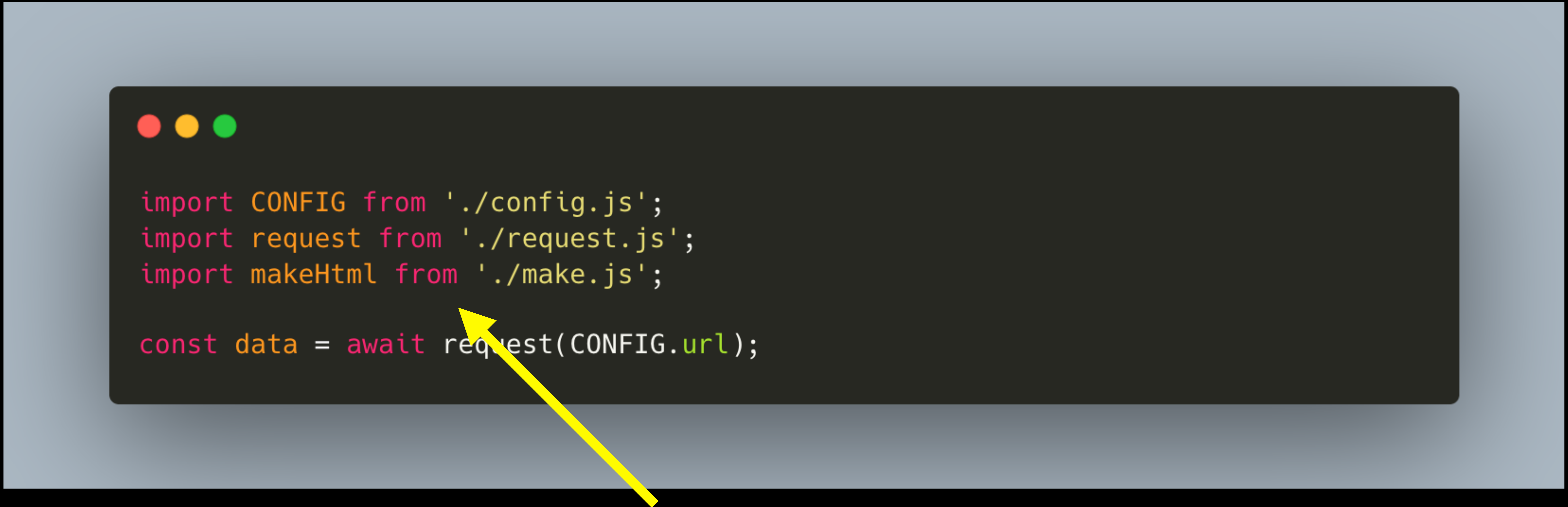
Modules



```
async function request(url) {
    let res = await fetch(url);
    return await res.json();
}

export default request
```

export default whaaaat?

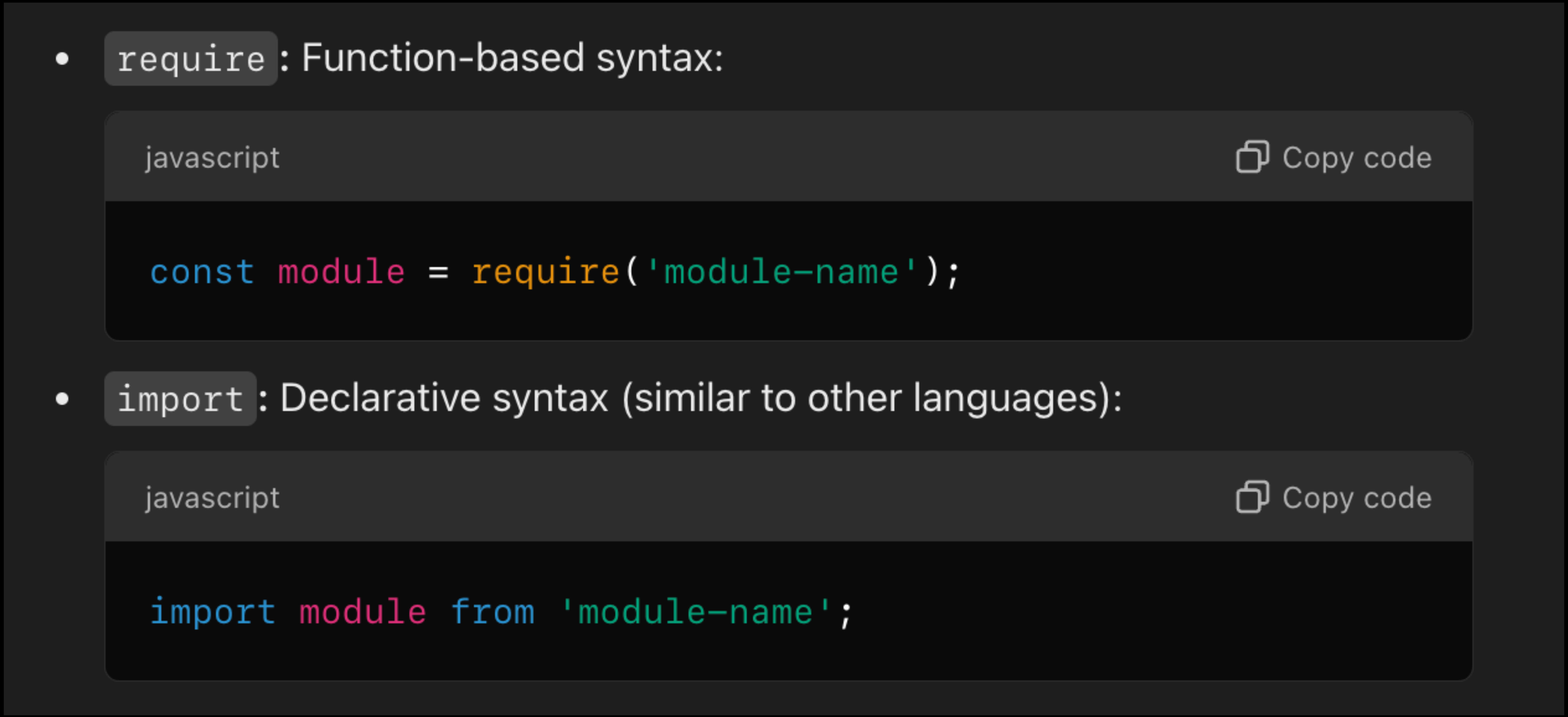
Modules



```
import CONFIG from './config.js';
import request from './request.js';
import makeHtml from './make.js';

const data = await request(CONFIG.url);
```

Import default whaaaat?

Modules



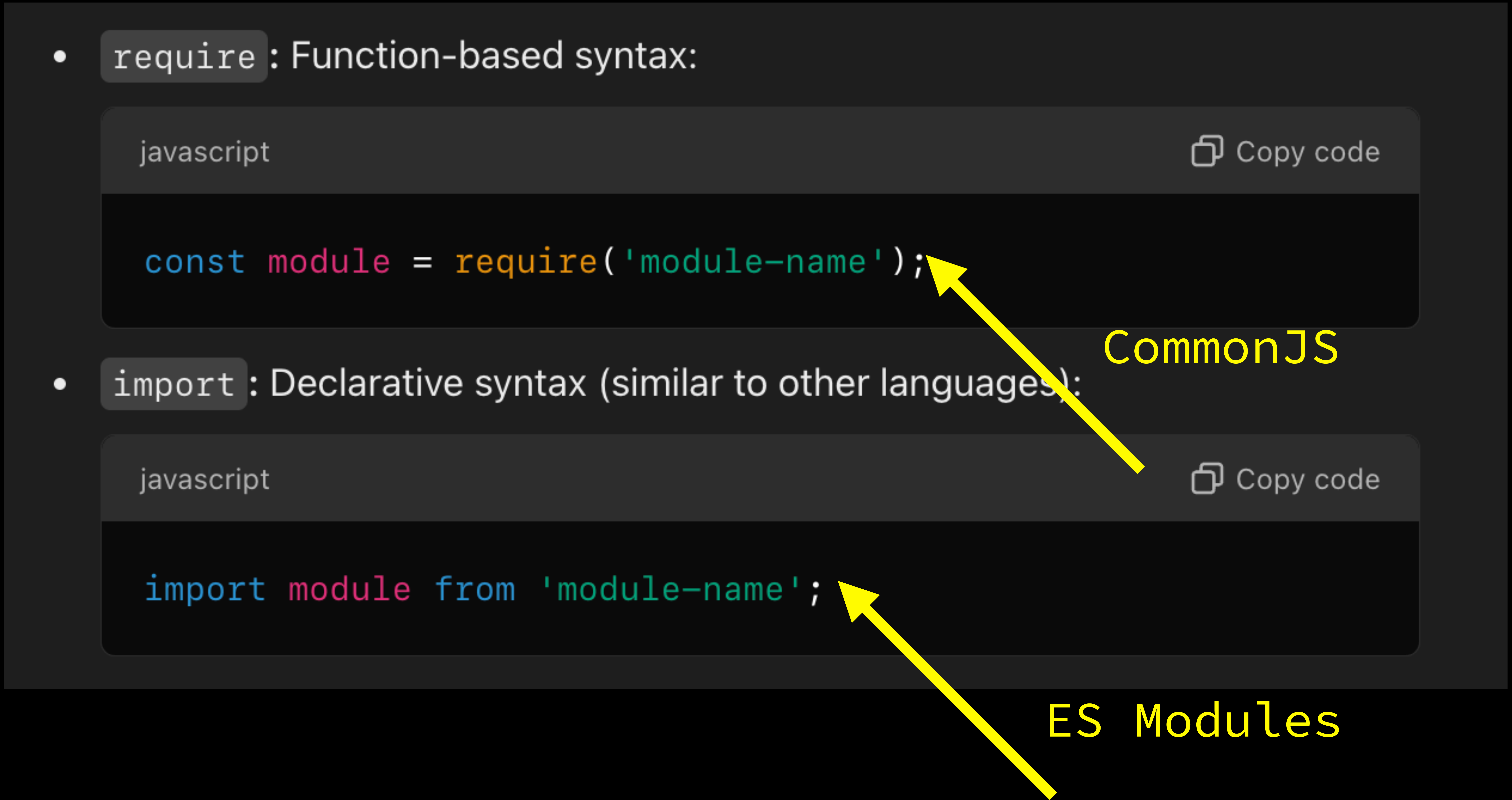
- `require`: Function-based syntax:

```javascript
const module = require('module-name');
```

- `import`: Declarative syntax (similar to other languages):

```javascript
import module from 'module-name';
```

Modules



- `require`: Function-based syntax:

javascript Copy code

```javascript
const module = require('module-name');
```

- `import`: Declarative syntax (similar to other languages):

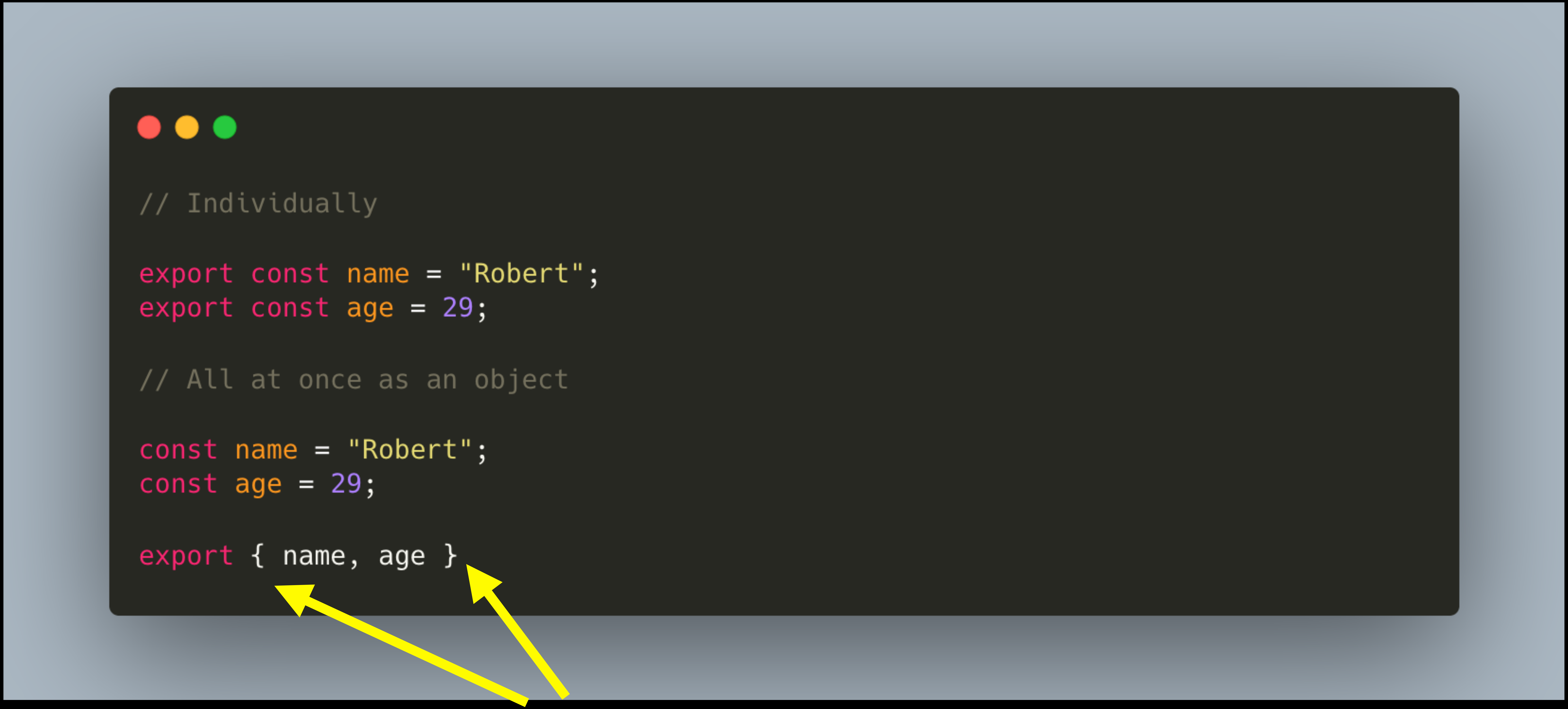
javascript Copy code

```javascript
import module from 'module-name';
```

Modules

- You can export a function or variable from any file
- There are two types of exports, named and default

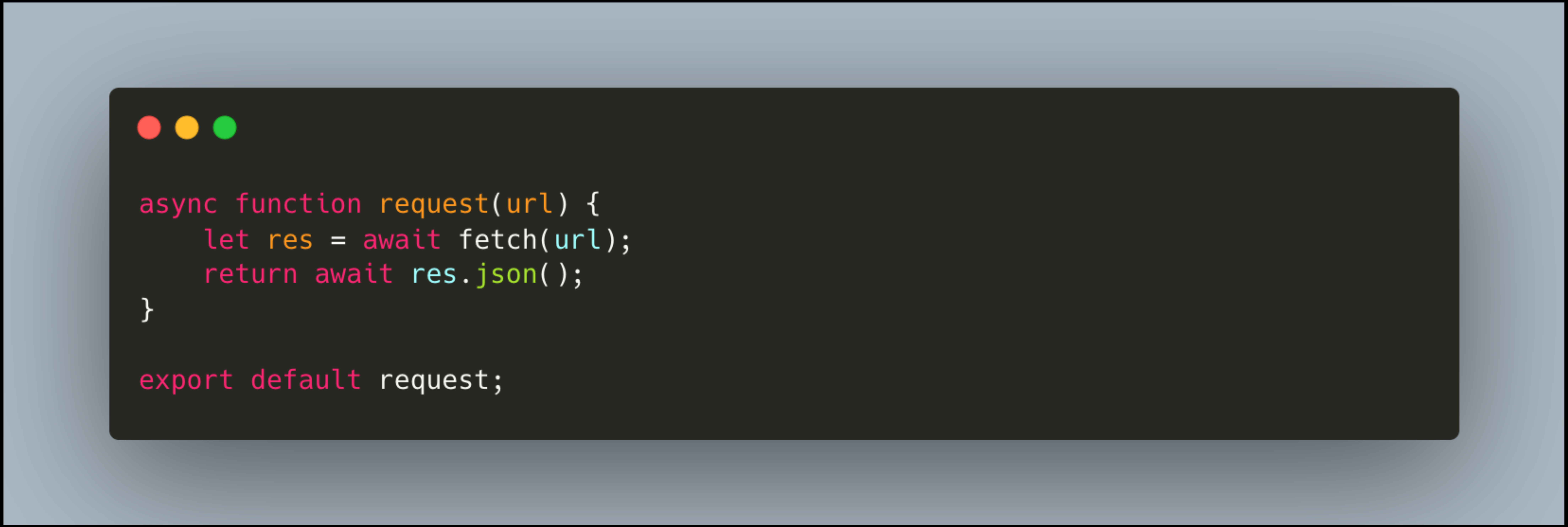
Named exports



```
// Individually

export const name = "Robert";
export const age = 29;

// All at once as an object

const name = "Robert";
const age = 29;

export { name, age }
```

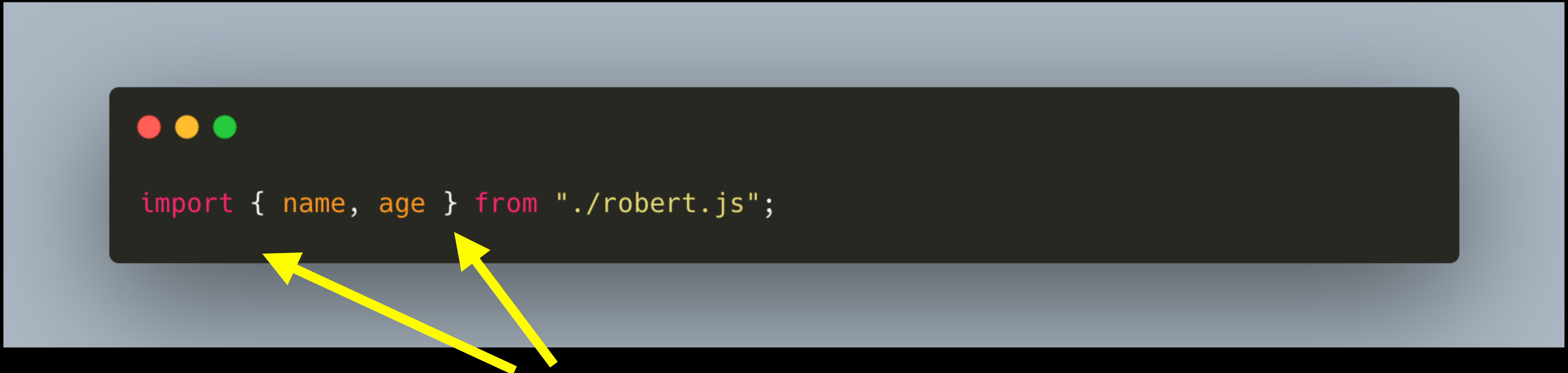
Exporting as an object, due to the {}

Default exports



```js
async function request(url) {
    let res = await fetch(url);
    return await res.json();
}

export default request;
```

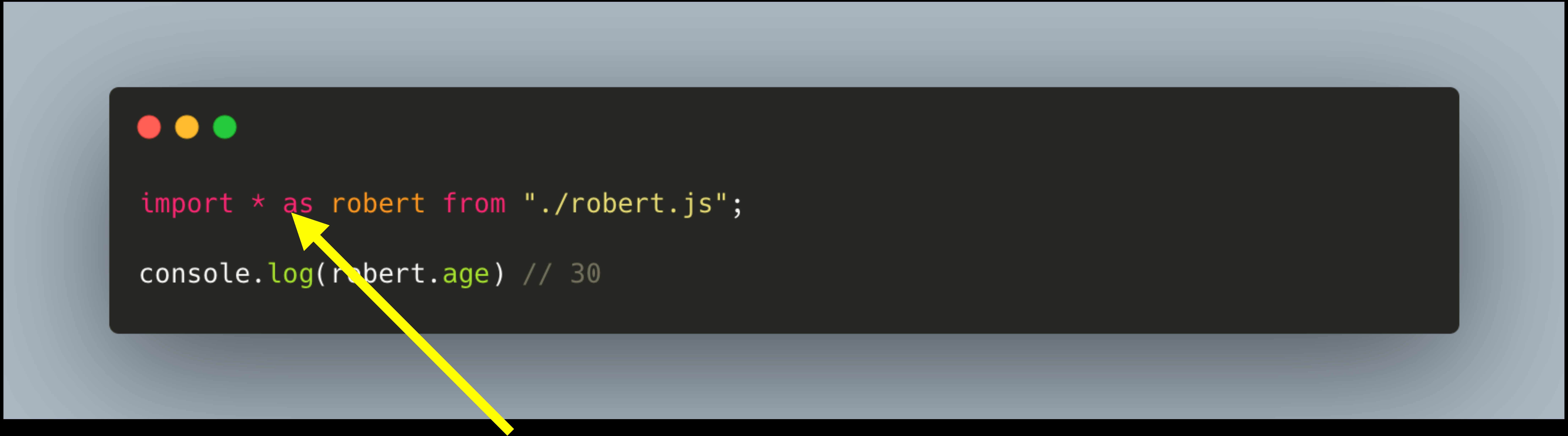
Named imports



```
import { name, age } from "./robert.js";
```

Importing as an object, due to the {}

Named imports



```
import * as robert from "./robert.js";

console.log(robert.age) // 30
```

Import everything using “as” to name it

Modules = functional programming

Referential transparency: The function always gives the same return value for the same arguments. This means that the function cannot depend on any mutable state

Side-effect free: The function cannot cause any side effects. Side effects may include I/O (e.g., writing to the console or a log file), modifying a mutable object, reassigning a variable, etc.

Why?

- Using modules allow us to work in components
- Working in components allows us to:
  - Re-use snippets of code (DRY)
  - Write cleaner code (KISS)
  - Debug with more ease instead of 99999 lines of file xyz
- Prepares us to work with external modules (see Svelte, see D3)

Aan de slag

Maak de *namedExports* opdracht.

Aan de slag

Maak de *dataFetch* opdracht.

Aan de slag

Installeer *D3* in je project.

Schedule

1. Review assignments (hof, scope, pure)
2. Modules (import / export)
3. **Dataset research**
4. All together!

Dataset

We gaan aan de slag met het zoeken van een dataset die je wilt gaan visualiseren voor je individuele eindopdracht.

Kies een onderwerp wat je zelf interessant lijkt. Volgende week *Maandag een* **show en tell**. *Donderdag* lever je een **concept idee** in.

Dataset eisen

- Je hebt een dataset die **enigszins dynamisch** is
- Je hebt een dataset die **enigszins groot** is (+10.000 punten)
- Je hebt een dataset met **query parameters en filter opties**
- Je hebt een dataset die met **gangbare formaten (.json) werkt**
- Je hebt een dataset die **blijft werken tot het einde** 🙃

Dataset kiezen

Maar belangrijker; een dataset waar je een **onderzoeksvraag uit kan halen**.

e.g. een patroon dat je kan visualiseren, een data vraag die je kan beantwoorden door vergelijken etc.

API overzichten

- https://rapidapi.com/
- https://publicapis.dev/
- https://github.com/public-apis/public-apis
- https://www.kaggle.com/

API instancies

- Gemeente: https://data.amsterdam.nl/
- Musea: http://data.rijksmuseum.nl/
- Overheden: https://data.gov/

Volgende les Show en Tell

- **Ga opzoek naar een goede dataset**
- Waarom deze dataset? Waarom dit onderwerp?
- Wat is je onderzoeksvraag bij de dataset?
- Hoe is de documentatie van de API?
- Is de dataset dynamisch en up-to-date?
- Welke formaten geeft de API terug qua data?

```
Uncaught SyntaxError
Unexpected end of input
```