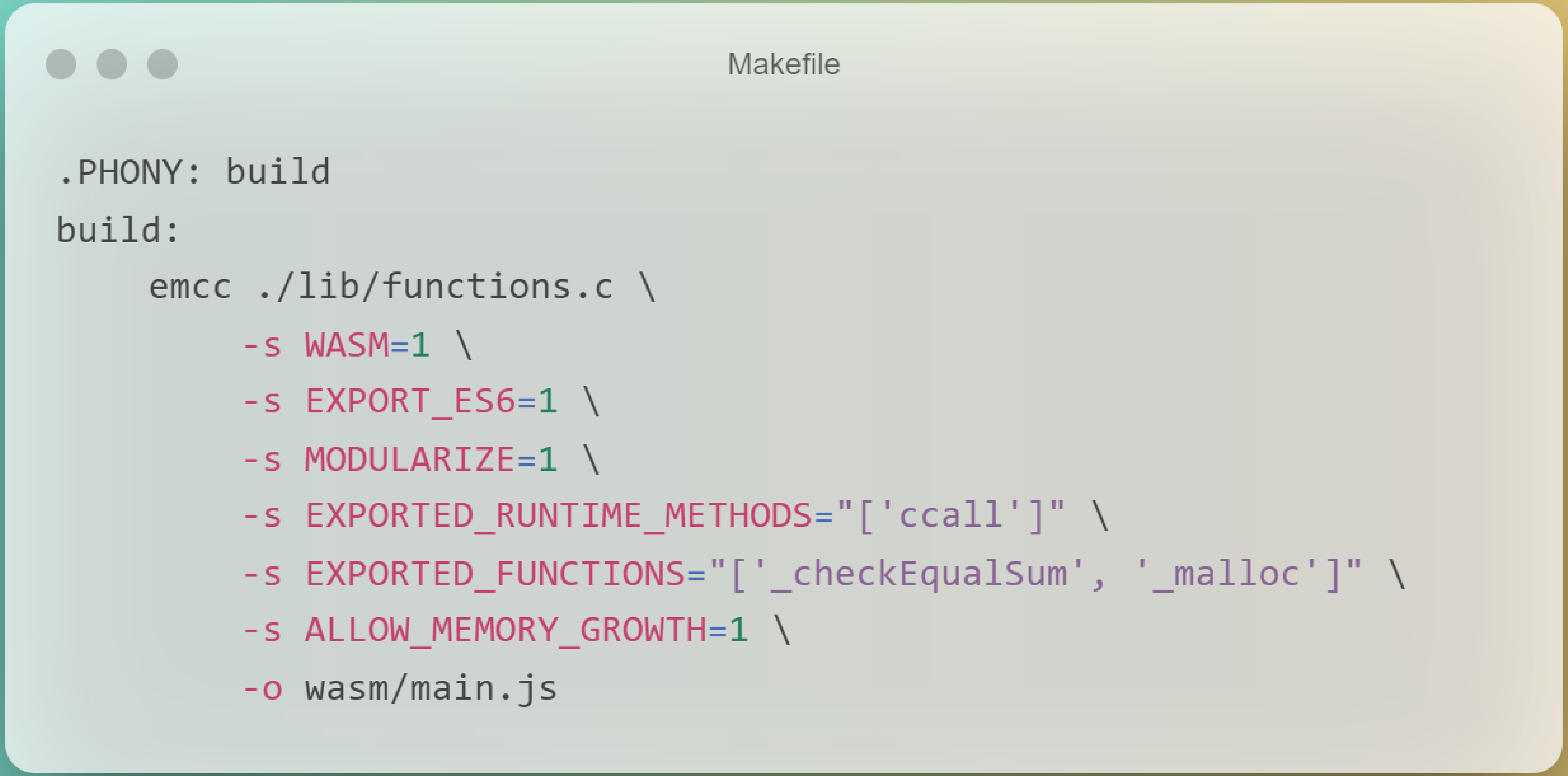
WebAssembly

Grupo 2

C => WASM + JS

C => WASM + JS

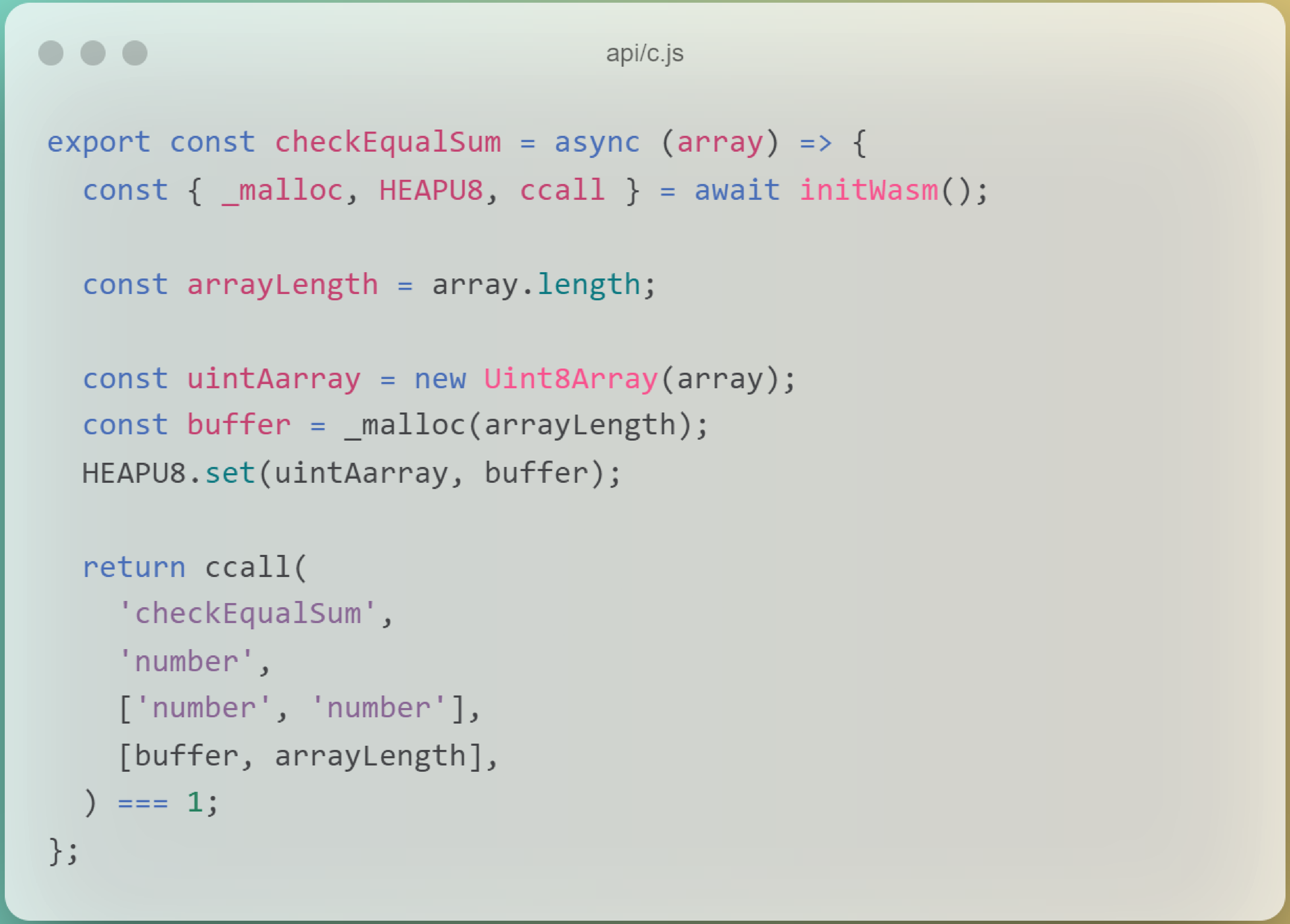
EMScripten


Makefile

```
.PHONY: build
build:
    emcc ./lib/functions.c \
        -s WASM=1 \
        -s EXPORT_ES6=1 \
        -s MODULARIZE=1 \
        -s EXPORTED_RUNTIME_METHODS="['ccall']" \
        -s EXPORTED_FUNCTIONS="['_checkEqualSum', '_malloc']" \
        -s ALLOW_MEMORY_GROWTH=1 \
        -o wasm/main.js
```

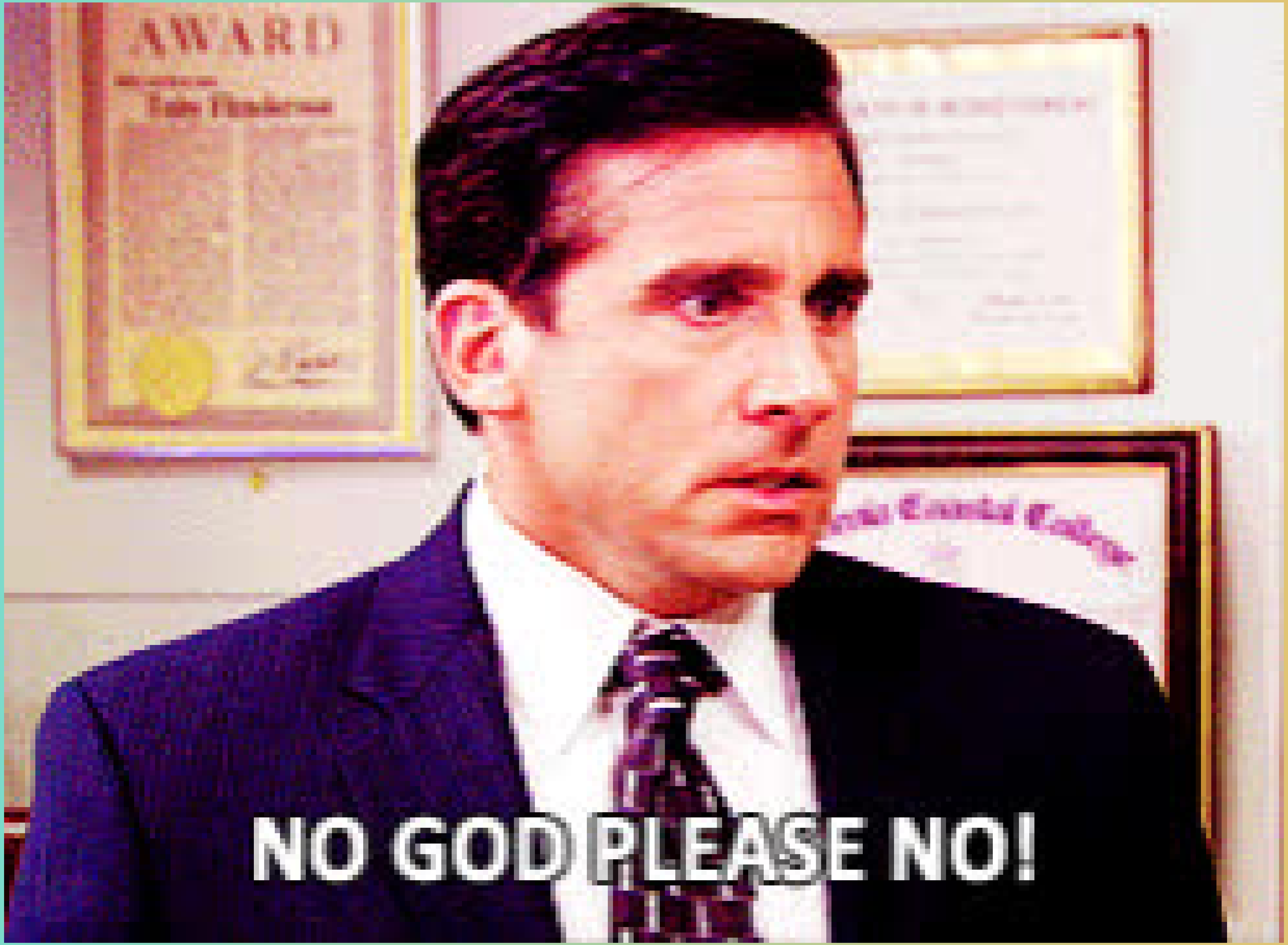
WASM + JS => app

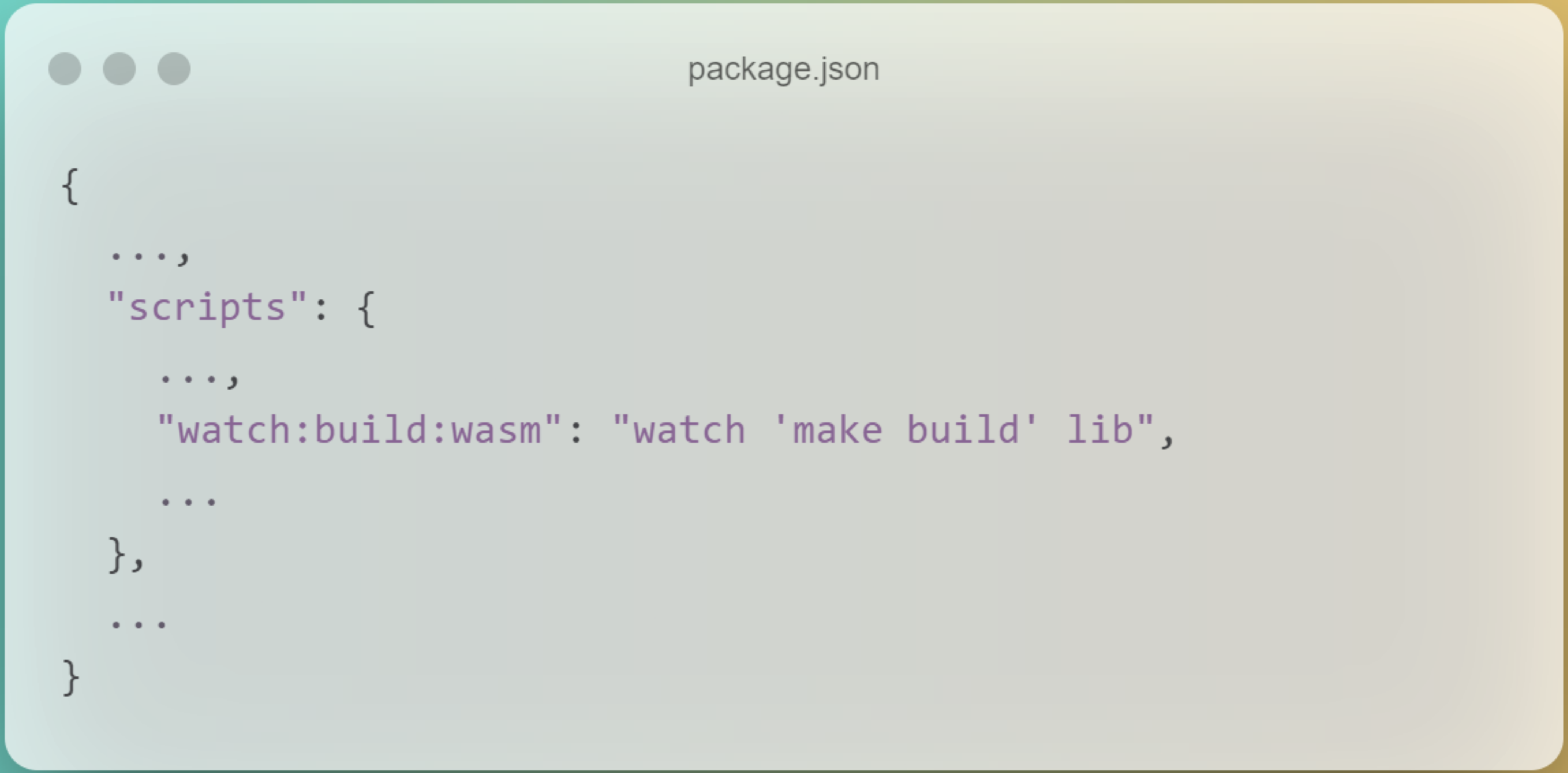
??


api/c.js

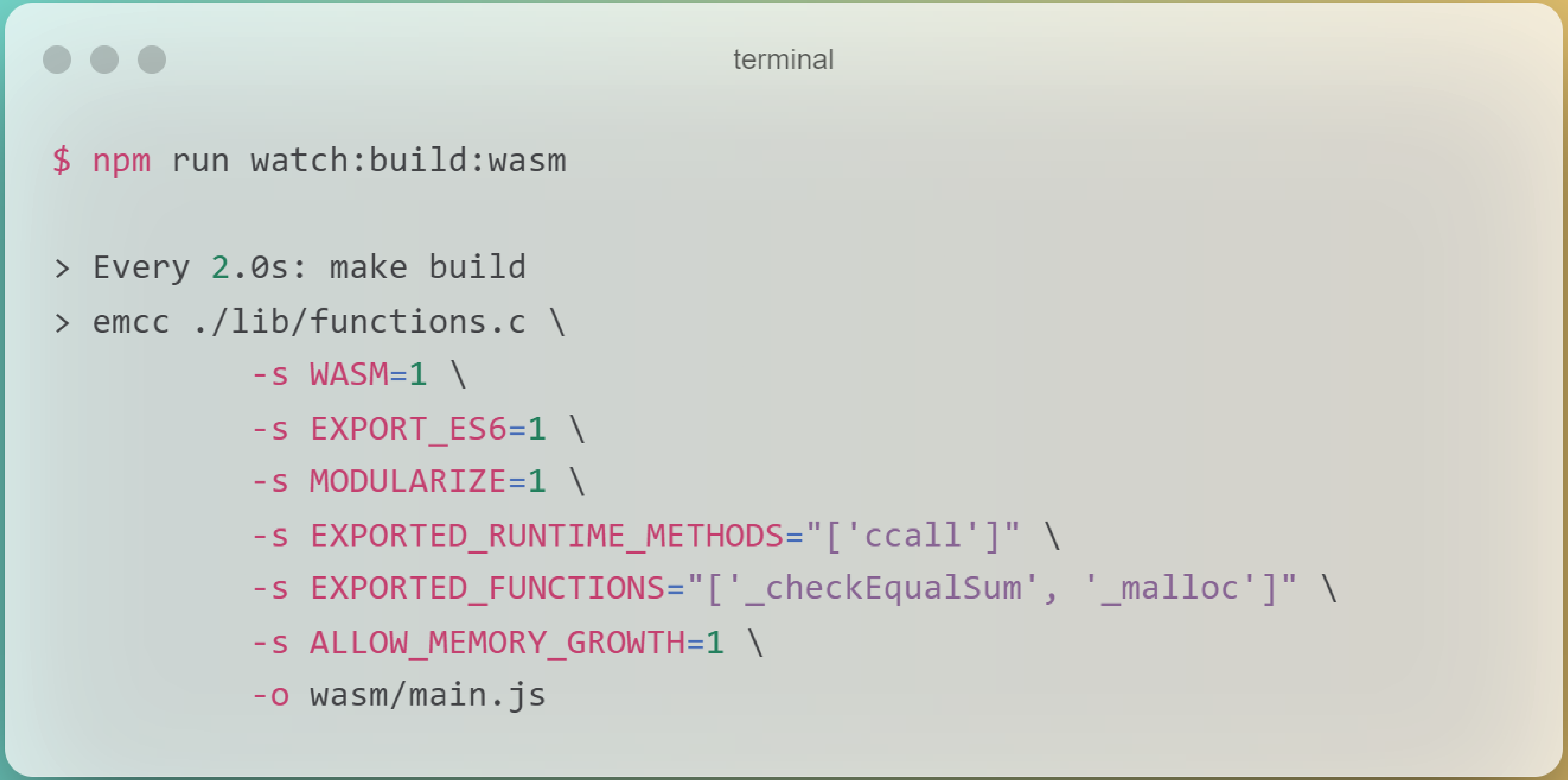
```
export const checkEqualSum = async (array) => {
  const { _malloc, HEAPU8, ccall } = await initWasm();

  const arrayLength = array.length;

  const uintAarray = new Uint8Array(array);
  const buffer = _malloc(arrayLength);
  HEAPU8.set(uintAarray, buffer);

  return ccall(
    'checkEqualSum',
    'number',
    ['number', 'number'],
    [buffer, arrayLength],
  ) === 1;
};
```

Flujo de trabajo


AWARD
NO GOD PLEASE NO!


package.json

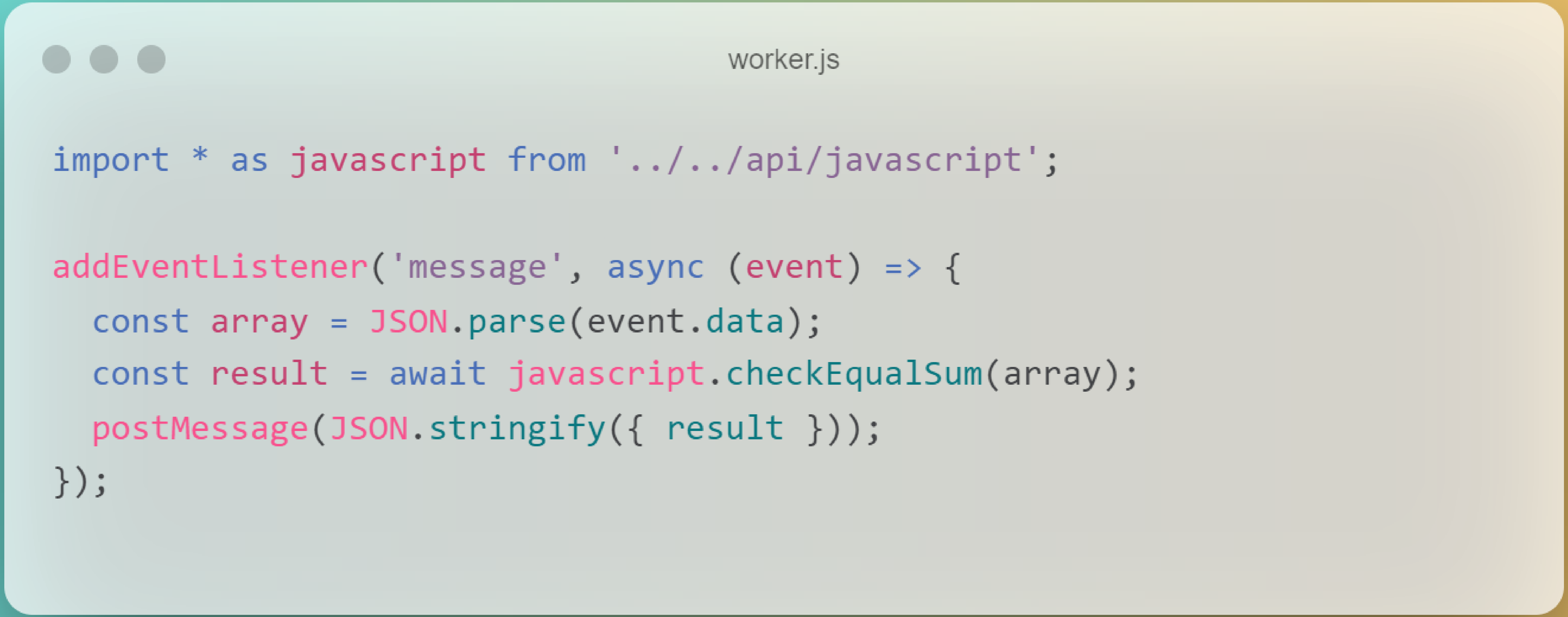
```json
{
  ...,
  "scripts": {
    ...,
    "watch:build:wasm": "watch 'make build' lib",
    ...
  },
  ...
}
```



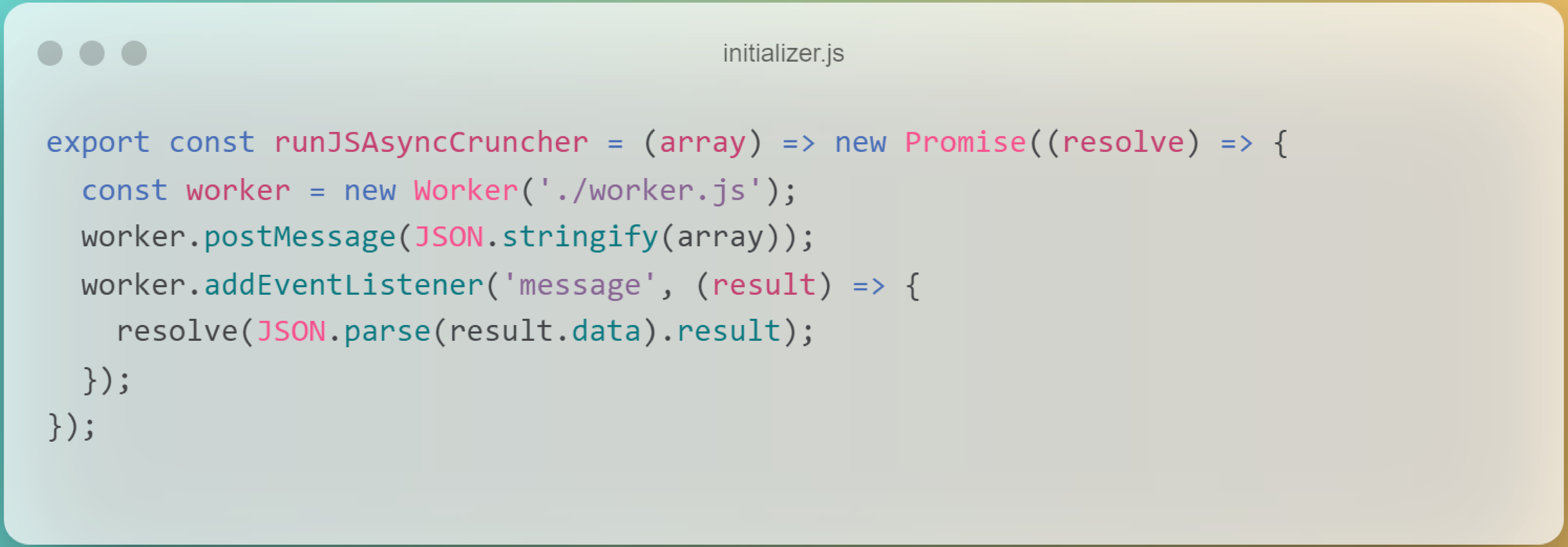
```
$ npm run watch:build:wasm

> Every 2.0s: make build
> emcc ./lib/functions.c \
          -s WASM=1 \
          -s EXPORT_ES6=1 \
          -s MODULARIZE=1 \
          -s EXPORTED_RUNTIME_METHODS="['ccall']" \
          -s EXPORTED_FUNCTIONS="['_checkEqualSum', '_malloc']" \
          -s ALLOW_MEMORY_GROWTH=1 \
          -o wasm/main.js
```

WASM !== Solución

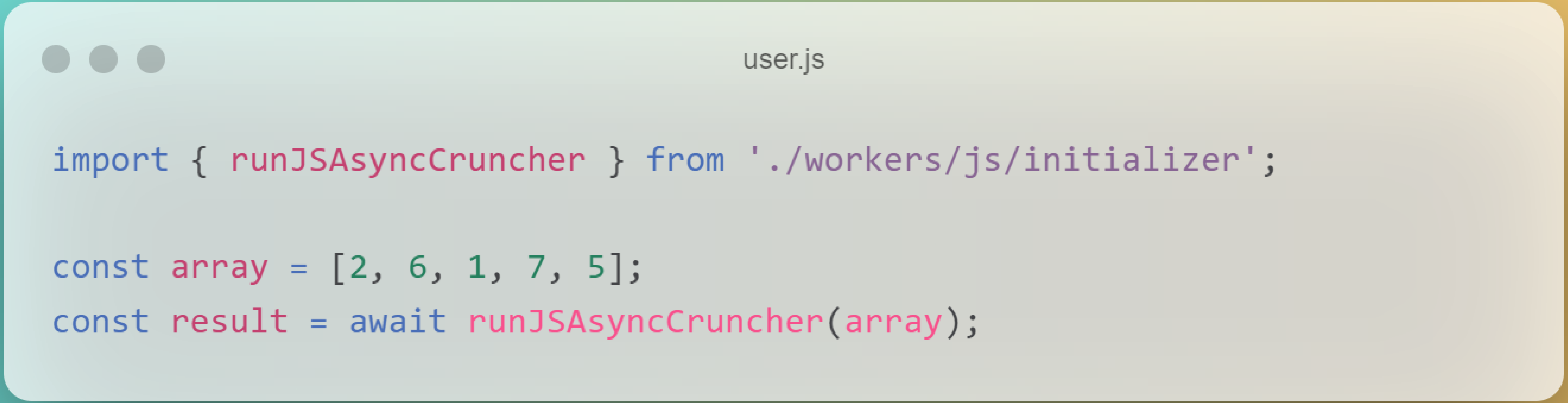
El usuario tratando de
interactuar con la página
JavaScript



```javascript
import * as javascript from '../../api/javascript';

addEventListener('message', async (event) => {
  const array = JSON.parse(event.data);
  const result = await javascript.checkEqualSum(array);
  postMessage(JSON.stringify({ result }));
});
```



```
export const runJSAsyncCruncher = (array) => new Promise((resolve) => {
  const worker = new Worker('./worker.js');
  worker.postMessage(JSON.stringify(array));
  worker.addEventListener('message', (result) => {
    resolve(JSON.parse(result.data).result);
  });
});
```


user.js

```
import { runJSAsyncCruncher } from './workers/js/initializer';

const array = [2, 6, 1, 7, 5];
const result = await runJSAsyncCruncher(array);
```