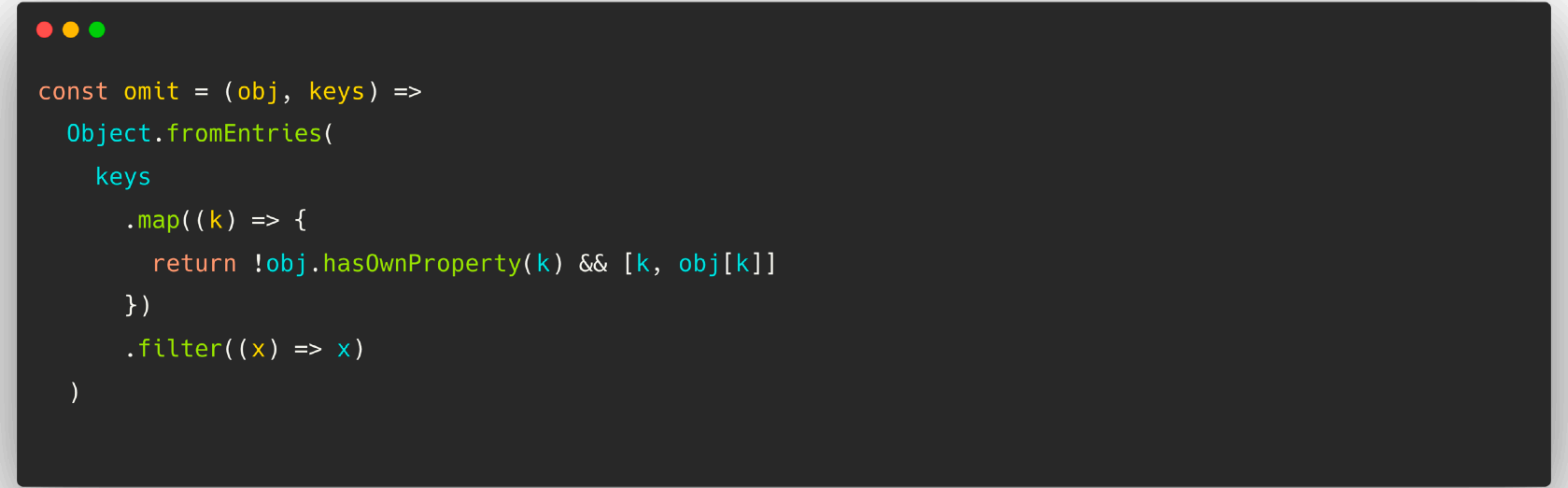
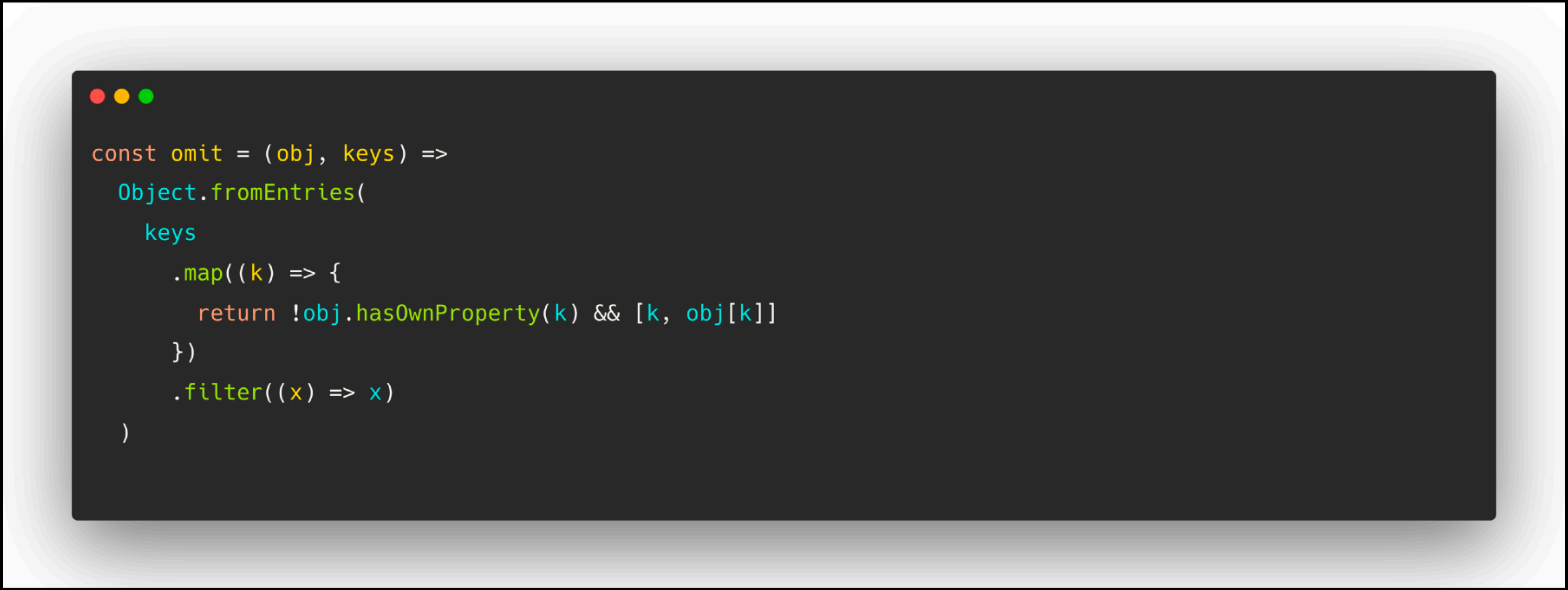


```
const omit = (obj, keys) =>
  Object.fromEntries(
    keys
      .map((k) => {
        return !obj.hasOwnProperty(k) && [k, obj[k]]
      })
      .filter((x) => x)
  )
```



```
const omit = (obj, keys) =>
  Object.fromEntries(
    keys
      .map((k) => {
        return !obj.hasOwnProperty(k) && [k, obj[k]]
      })
      .filter((x) => x)
  )
```

The major challenges

- **It is not ergonomic!** 🥸

- **Destructuring cannot:** 👎

  pick/omit properties dynamically (requires a hardcoded number of properties)

  pick up properties from the prototype

  omit some properties, and we can only clone and delete