Contents

The state of JavaScript

Today, more developers use JavaScript than any other language in the world. It is the common tongue that ties together different ways of thinking, and programming. However, most books either

1. Dwell purely on the syntax of the language, ignoring the rationale and development of several *approaches* of developing
2. Focus on a specific paradigm.

This is a companion book. It introduces all the paradigms and traces the evolution of JavaScript, as well as its architectural foundation.

We will go over

1. JavaScript's development
2. The different and distinct communities using the language
3. Mental models each community has developed separately
4. Common patterns to encounter.

Why read this book? By understanding each "branch" of the JavaScript family, you become a versatile user of all branches. You will understand the different constraints guiding the evolution of this language.

Early days

JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2, and became the ECMA-262 standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla Foundation continued to develop JavaScript for the Firefox browser. Mozilla's latest version was 1.8.5.

Node.js

Node.js is a free, open-source, cross-platform JS *runtime environment.* It was written by Ryan Dahl in 2009.

A runtime lets JS run on the computer; Dahl wanted to write servers. He boasted Node.js's optimizations when it came to concurrent connections and asynchronous programming. The project combined the Google V8 engine (used in Chromium), an event loop, and a low-level I/O API that could be called in JS.

The main reason for Node's adoption turned out to be its package manager, npm: the Node Package Manager. It allowed developers to publish and share Node packages and source code and use each other's code very easily. Analogous to Tex's CTAN, PHP's composer, and Python's pip, npm differed in its scope. It was backed by Microsoft's GitHub and had the largest download rates ever. Even today, when developers speak of "supply-chain security", they're talking about node packages; most of the time.

This is Node.js's default project:

```javascript
const { createServer } = require('node:http');

const hostname = '127.0.0.1';
const port = 3000;

const server = createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World');
});

server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magnam aliquam quaerat voluptatem. Ut enim aeque doleamus animo, cum corpore dolemus, fieri tamen permagna accessio potest, si aliquod aeternum et infinitum impendere malum nobis opinemur. Quod idem licet transferre in voluptatem, ut postea variari voluptas distinguique possit, augeri amplificarique non possit. At etiam Athenis, ut e patre audiebam facete et urbane Stoicos irridente, statua est in quo a nobis philosophia defensa et collaudata est, cum id, quod maxime placeat, facere possimus, omnis voluptas assumenda est, omnis dolor repellendus. Temporibus autem quibusdam et aut officiis debitis aut rerum necessitatibus saepe eveniet, ut et voluptates repudiandae sint et molestiae non recusandae. Itaque earum rerum defuturum, quas natura non depravata desiderat. Et quem ad me accedis, saluto: 'chaere,' inquam, 'Tite!' lictores, turma omnis chorusque: 'chaere, Tite!' hinc hostis mi Albucius, hinc inimicus. Sed iure Mucius. Ego autem mirari satis non queo unde hoc sit tam insolens domesticarum rerum fastidium. Non est omnino hic docendi locus; sed ita prorsus existimo, neque eum Torquatum, qui hoc primus cognomen invenerit, aut torquem illum hosti detraxisse, ut aliquam ex eo est consecutus? – Laudem et caritatem, quae sunt vitae sine metu degendae praesidia firmissima. – Filium morte multavit. – Si sine causa, nollem me ab eo delectari, quod ista Platonis, Aristoteli, Theophrasti orationis ornamenta neglexerit. Nam illud quidem physici, credere aliquid esse minimum, quod profecto numquam putavisset, si a Polyaeno, familiari suo, geometrica discere maluisset quam illum etiam ipsum.