

# Node.js: The Big Picture

---

## CONSIDERING NODE.JS



**Paul O'Fallon**

@paulofallon



# Overview



How Node.js is commonly used

A brief history and technology origins

Some reasons *not* to use Node.js



# What is Node.js?

“As an asynchronous event driven JavaScript runtime, Node is designed to build scalable network applications.”



# How Did We Get Here?



Front-end Developer



Back-end Developer

# Considering Node as a Front-end Developer



Leverage your existing JavaScript skills

A back-end for Single Page Applications

Tooling for front-end developers



# Considering Node as a Back-end Developer



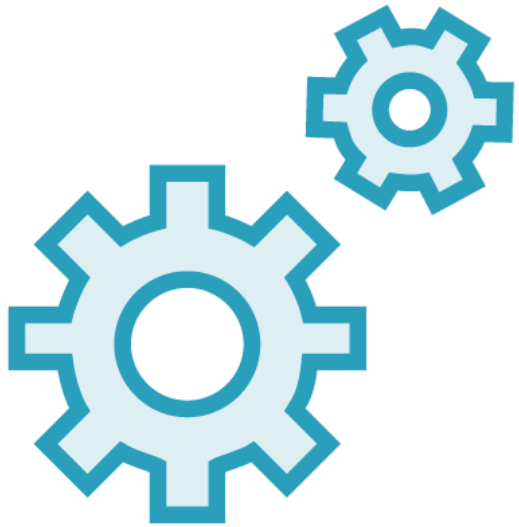
**An opportunity to learn JavaScript**

**Looking for a “lightweight” alternative**

**Vibrant ecosystem and platform support**



# Where Is Node.js Commonly Found?



Microservices  
and APIs



# Node.js for Microservices

```
const http = require('http');

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

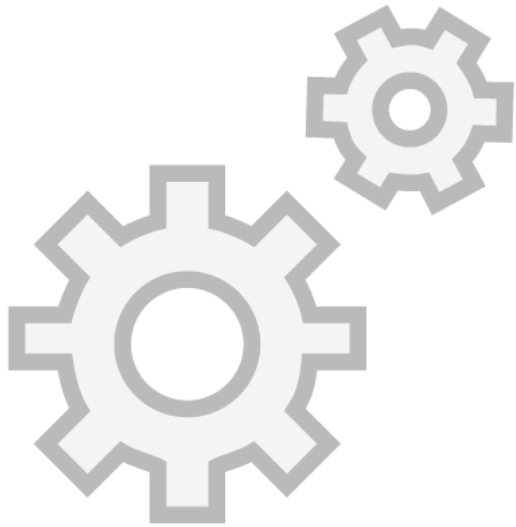
const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

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





# Where Is Node.js Commonly Found?

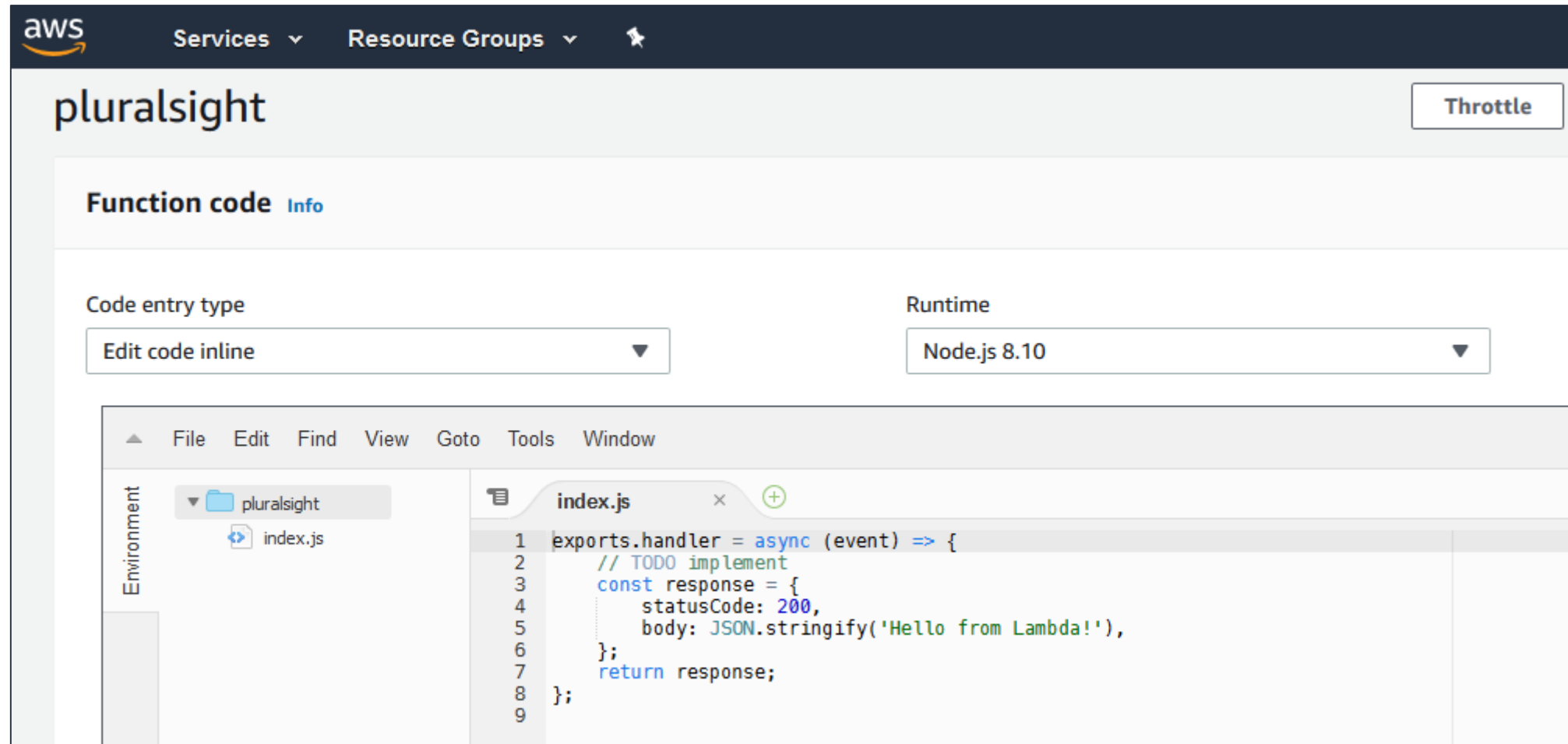


Microservices  
and APIs

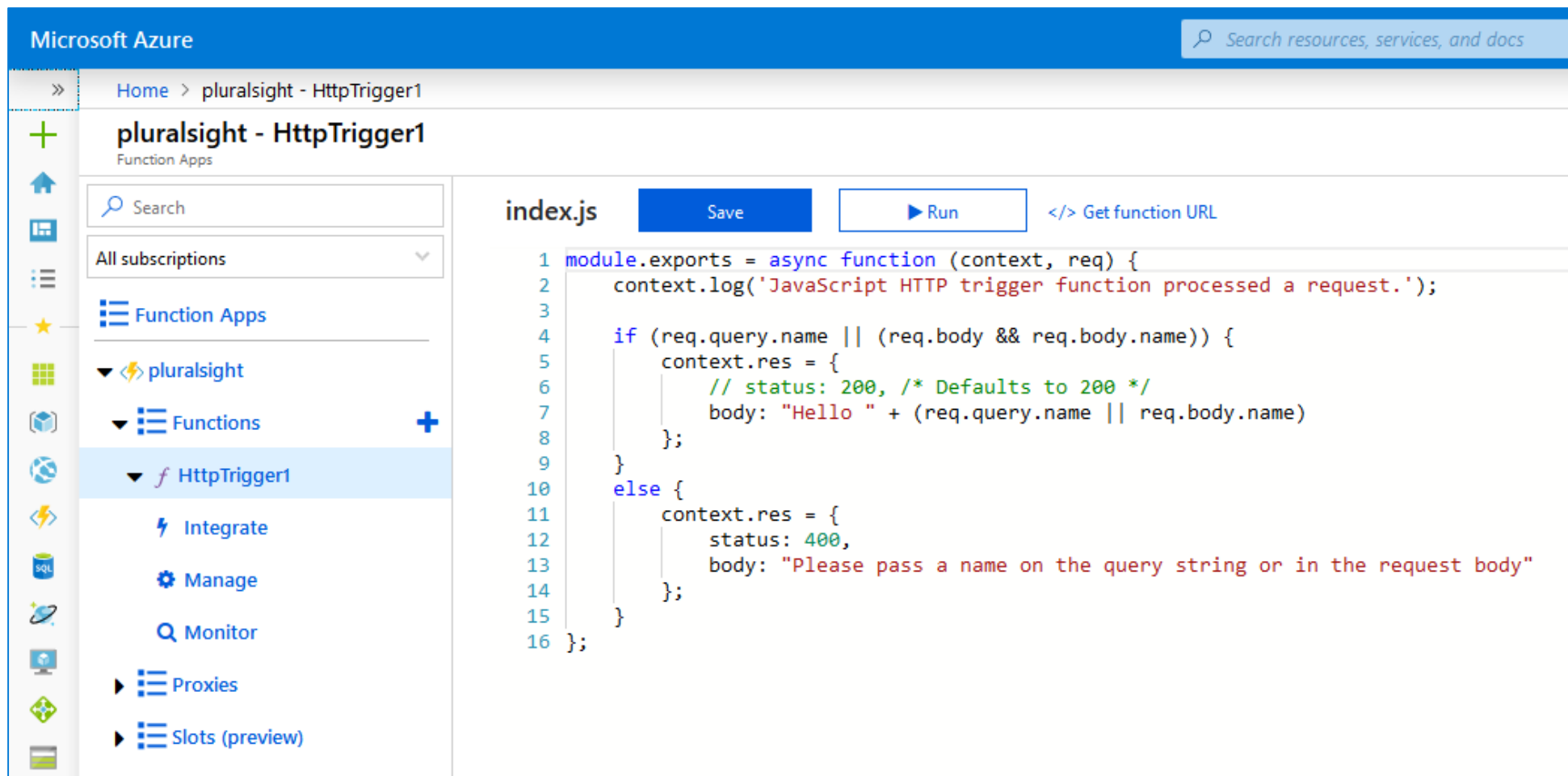


Serverless Cloud  
Functions

# Node.js for Serverless Functions in the Cloud



# Node.js for Serverless Functions in the Cloud



The screenshot displays the Microsoft Azure portal interface for a Function App named 'pluralsight - HttpTrigger1'. The left-hand navigation pane shows the hierarchy: Home > pluralsight - HttpTrigger1 > Functions > HttpTrigger1. The main area shows the 'index.js' file with the following code:

```
1 module.exports = async function (context, req) {
2   context.log('JavaScript HTTP trigger function processed a request.');
```

```
4   if (req.query.name || (req.body && req.body.name)) {
5     context.res = {
6       // status: 200, /* Defaults to 200 */
7       body: "Hello " + (req.query.name || req.body.name)
8     };
9   }
10  else {
11    context.res = {
12      status: 400,
13      body: "Please pass a name on the query string or in the request body"
14    };
15  }
16 };
```

Buttons for 'Save', 'Run', and '</> Get function URL' are visible above the code editor.



# Where Is Node.js Commonly Found?



Microservices  
and APIs

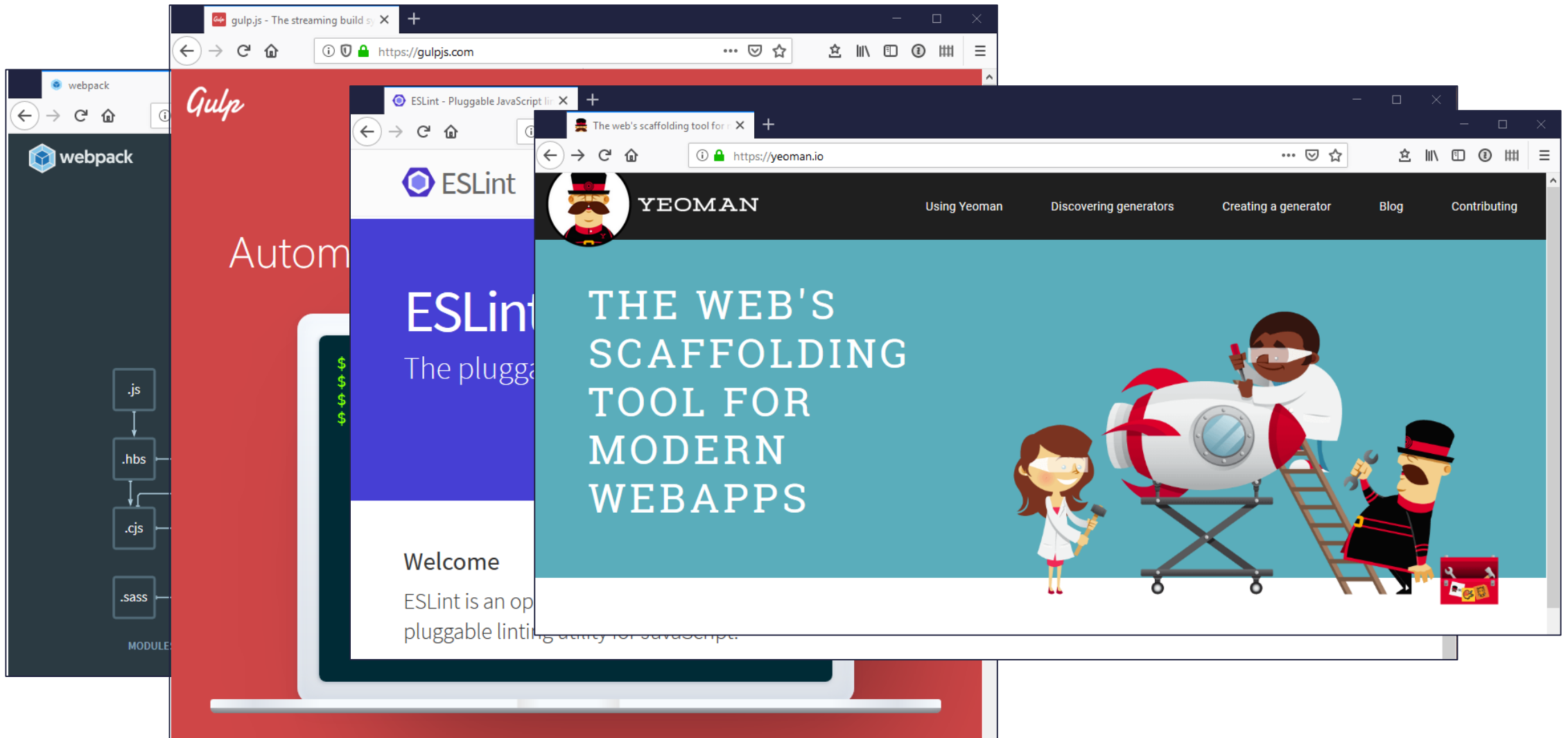


Serverless Cloud  
Functions



Command Line  
Applications

# Node.js for Command Line Apps



# Where Is Node.js Commonly Found?



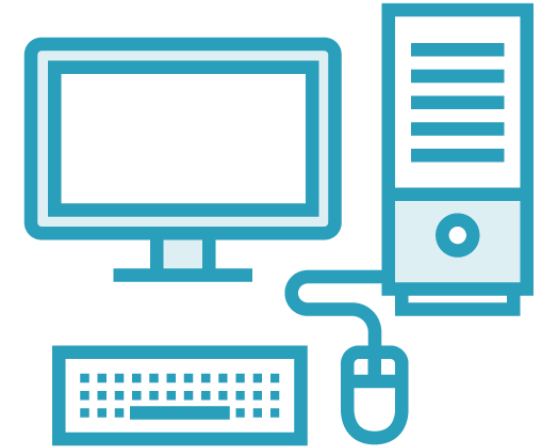
Microservices  
and APIs



Serverless Cloud  
Functions



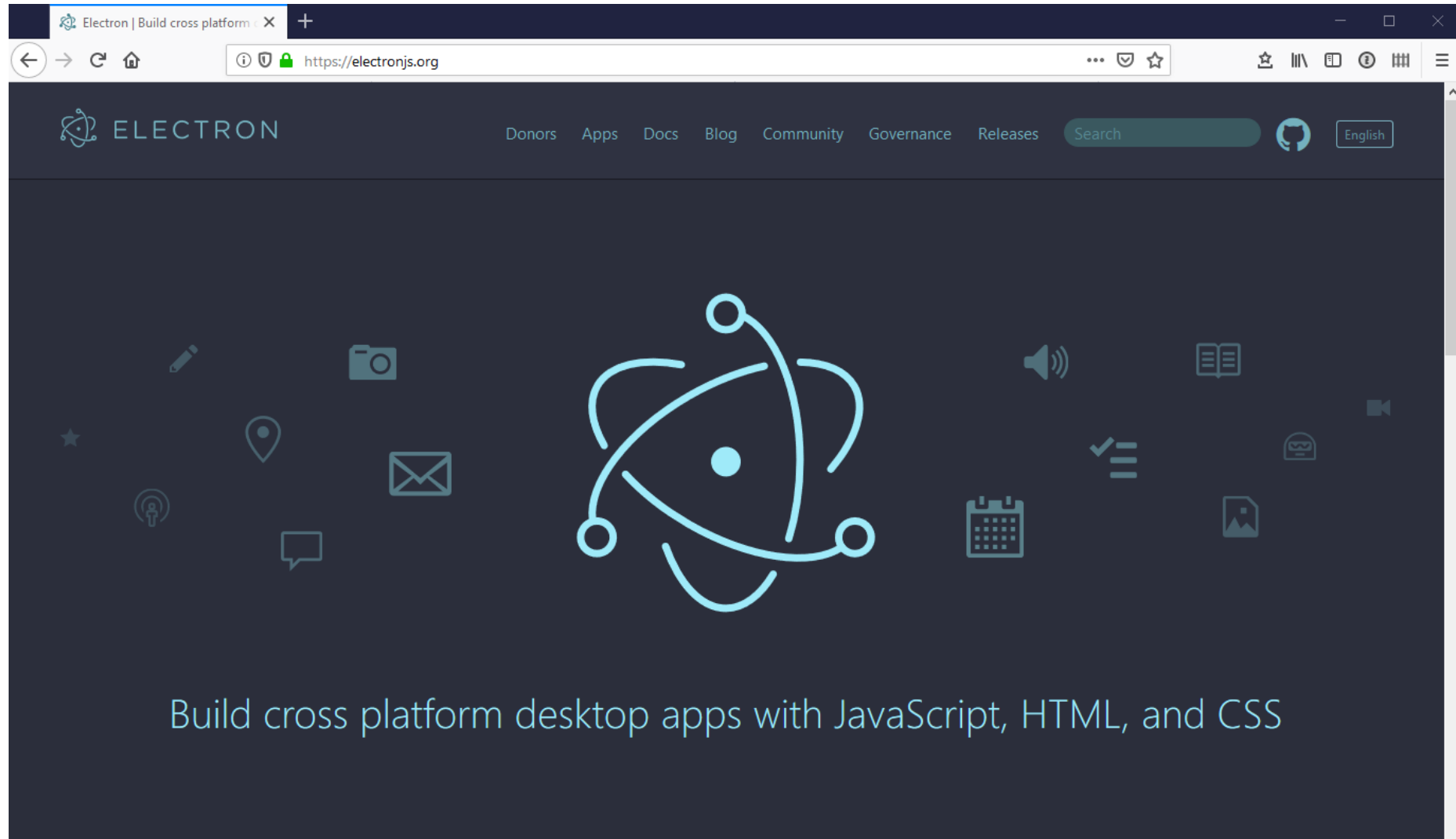
Command Line  
Applications



Desktop  
Applications



# Node.js for Desktop Applications



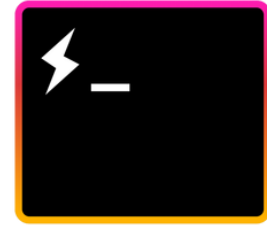
# Apps Built on Electron



Skype



GitHub Desktop



Hyper



Visual Studio Code



Slack



Atom

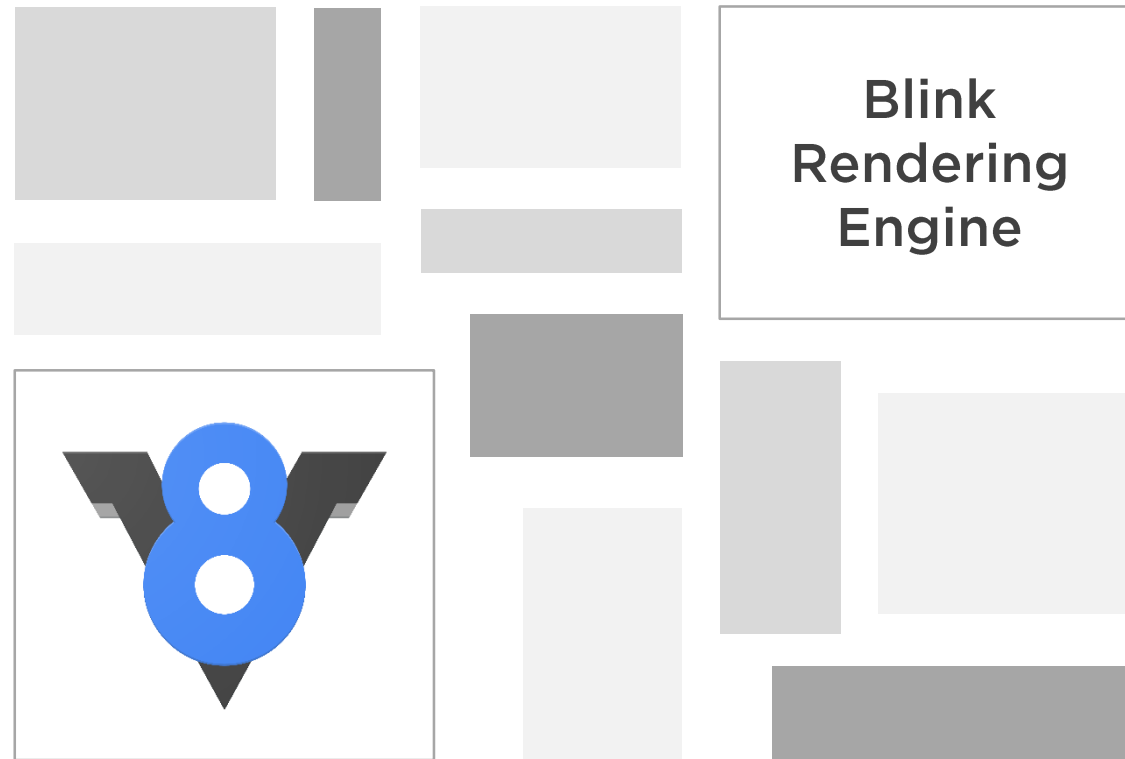


Atwood's Law:  
Any application that *can* be  
written in JavaScript,  
*will* eventually be written in  
JavaScript.



# What Makes up Node.js?

## Chrome browser



## Node.js

JavaScript code



C code

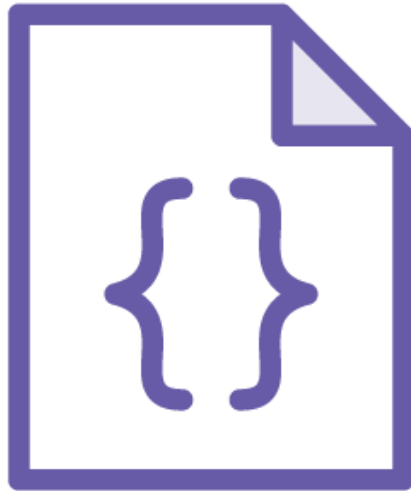


- JavaScript

- Asynchronous I/O
- Event Loop



# A Better Way to JavaScript



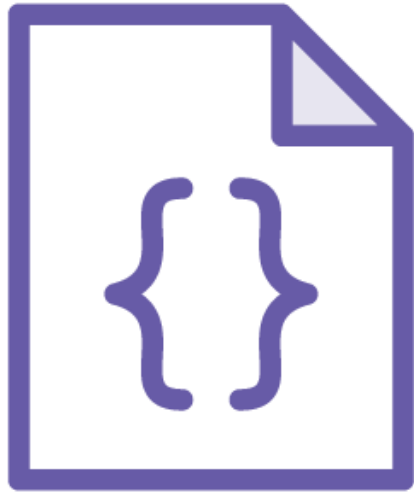
No more browser  
incompatibilities or  
polyfills!

Node.js ES2015/ES6, ES2016 and ...					
https://node.green					
Node.js ES2015 Support					
kangax's compat-table applied only to Node.js					
Nightly!					
13.0.0 12.1.0 11.15.0 10.15.3 10.8.0					
99% complete 99% complete 99% complete 99% complete 99% complete					
optimisation					
proper tail calls (tail call optimisation)					
direct recursion	?	Error	Error	Error	Error
mutual recursion	?	Error	Error	Error	Error
syntax					
default function parameters					
basic functionality	?	Yes	Yes	Yes	Yes
explicit undefined defers to the default	?	Yes	Yes	Yes	Yes
defaults can refer to previous params	?	Yes	Yes	Yes	Yes
arguments object interaction	?	Yes	Yes	Yes	Yes
temporal dead zone	?	Yes	Yes	Yes	Yes
separate scope	?	Yes	Yes	Yes	Yes
new Function() support	?	Yes	Yes	Yes	Yes

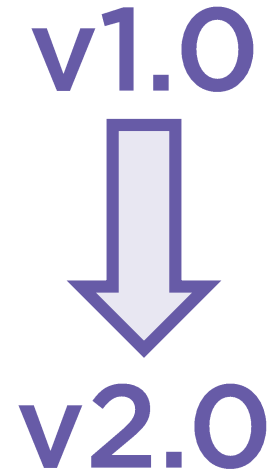


# A ~~Better~~ Way to JavaScript

Different



No more browser  
incompatibilities or  
polyfills!



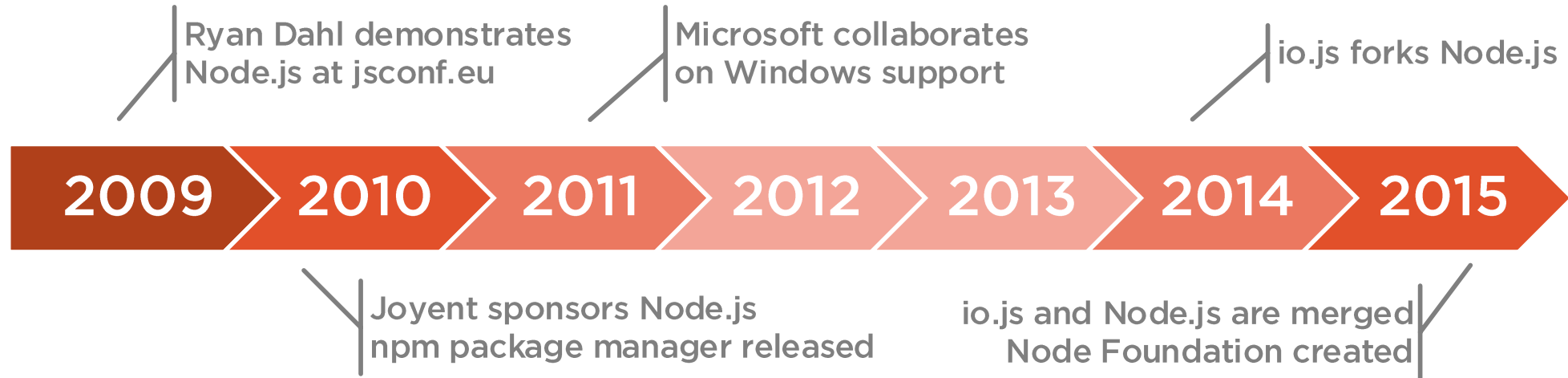
Adjust to changes in  
Node.js versions



Possibly OS-aware



# A Brief History of Node.js



# What Is the Release Cadence Today?



**New major versions released every 6 months**



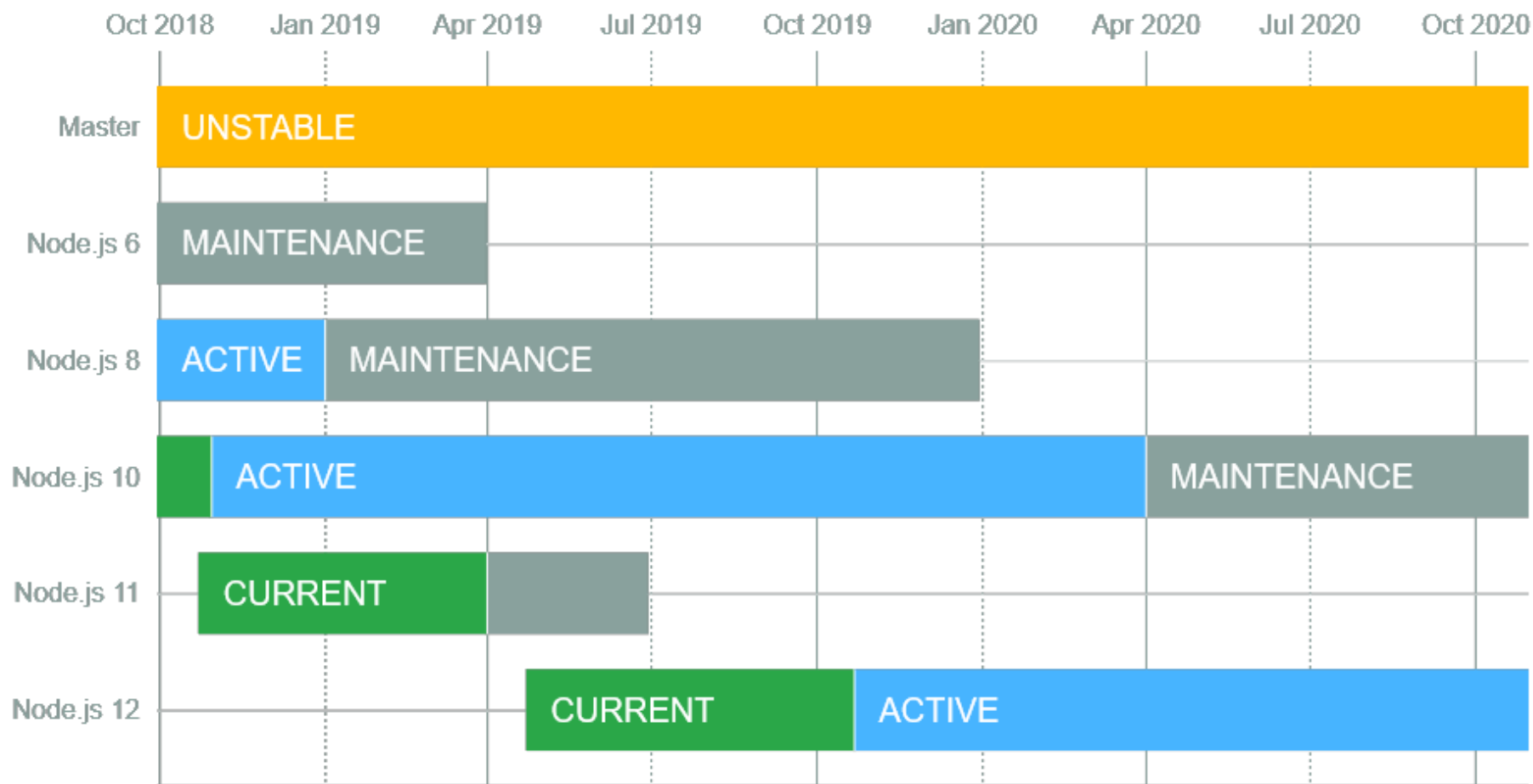
**Even-numbered major releases covered by long-term support (LTS)**



**No more than two active LTS releases at a time**



# What Is the Release Cadence Today?

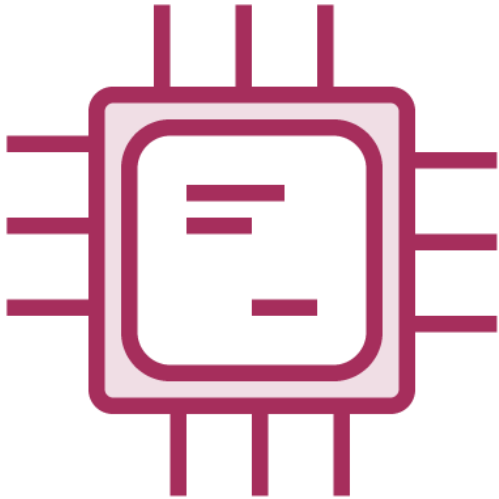


<https://github.com/nodejs/Release>





# Reasons Node.js May Not Be the Best Fit



You Have CPU-intensive Tasks



You Don't Like JavaScript