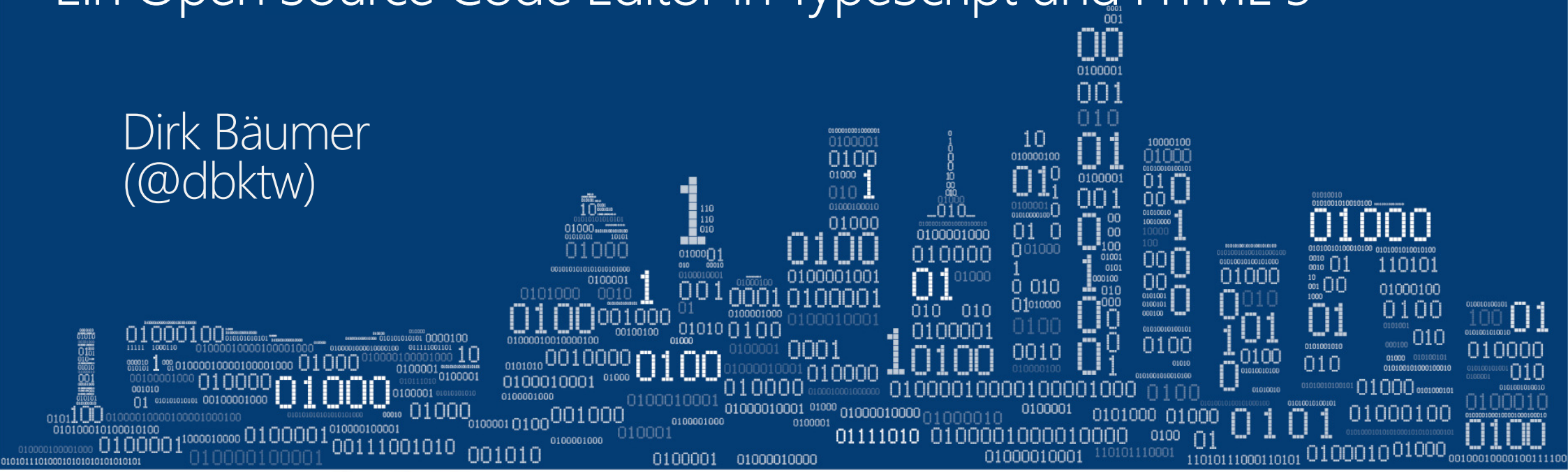
# Visual Studio Code

## Ein Open Source Code Editor in TypeScript und HTML 5

Dirk Bäumer
(@dbktw)



DIGICOMP

Microsoft

NOSER ENGINEERING
WE KNOW HOW

progile
productive not busy

trivadis
makes IT easier.

OFFERING SOLUTIONS SOFTWARE

# Visual Studio Code

A desktop tool that combines the **simplicity of a code editor** with what developers need for the core **code-build-debug-commit cycle**

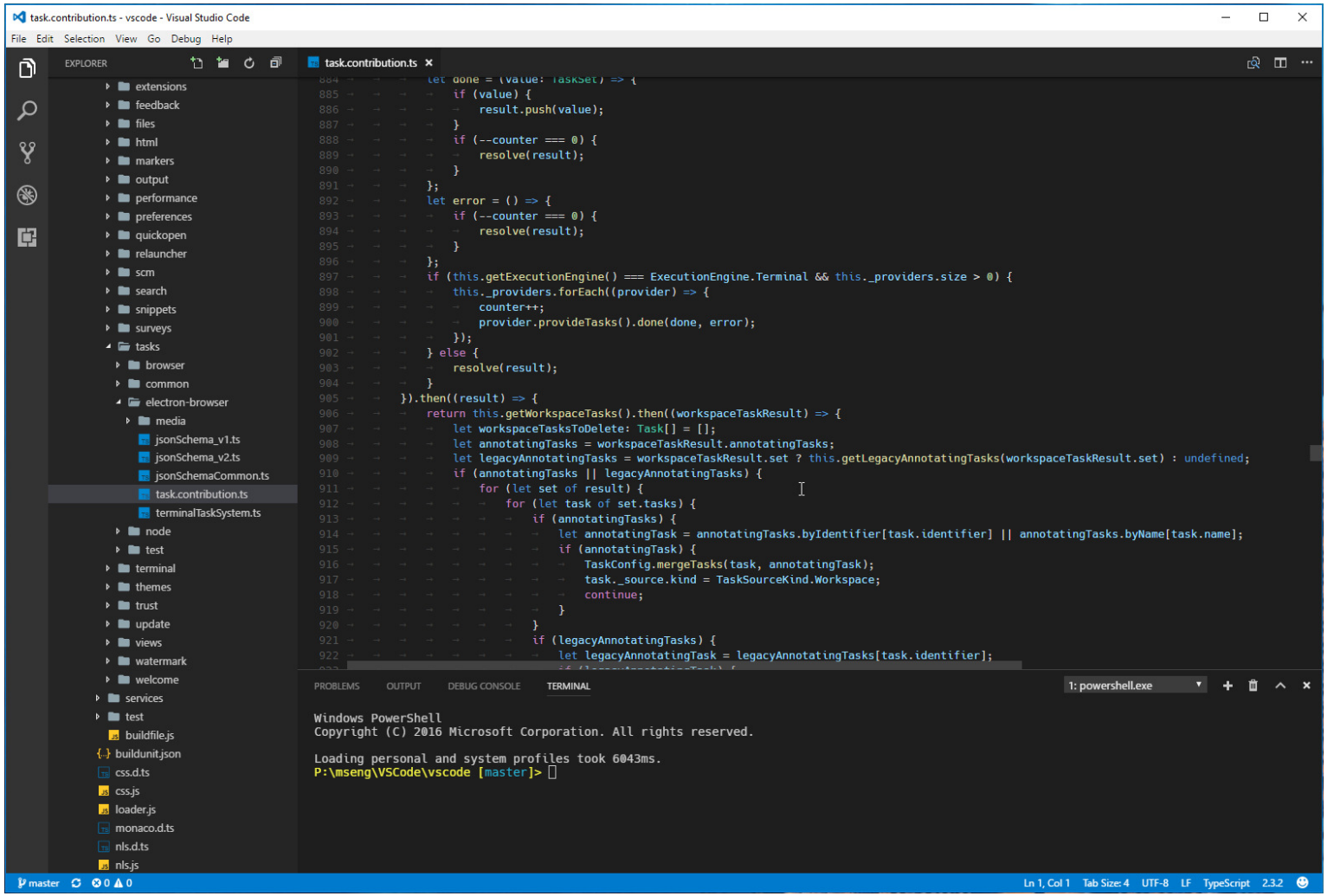
Windows
Windows 7, 8, 10

.deb
Debian, Ubuntu

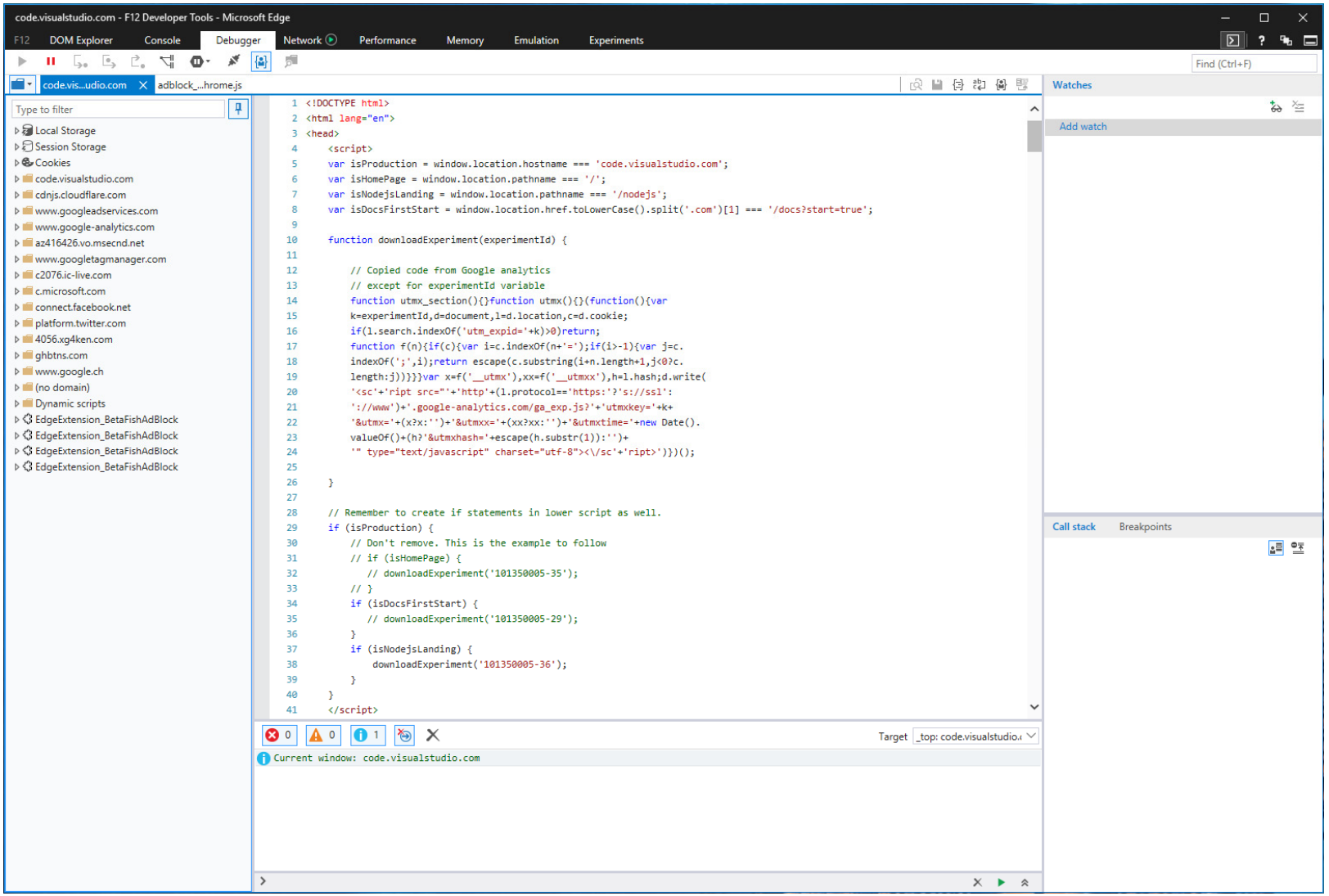
.rpm
Red Hat, Fedora, CentOS

OS X
OS X Yosemite, El Capitan

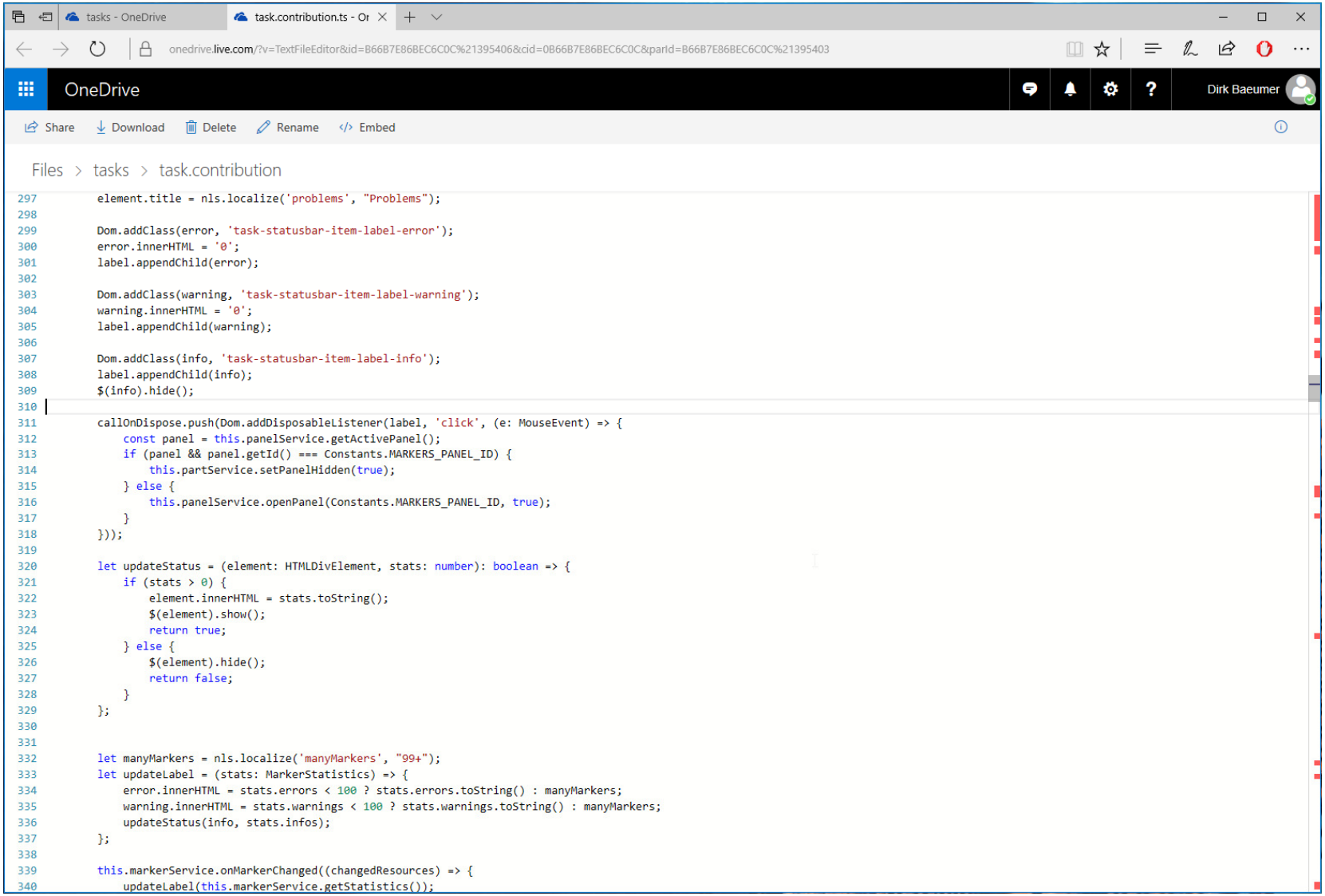
# Visual Studio Code



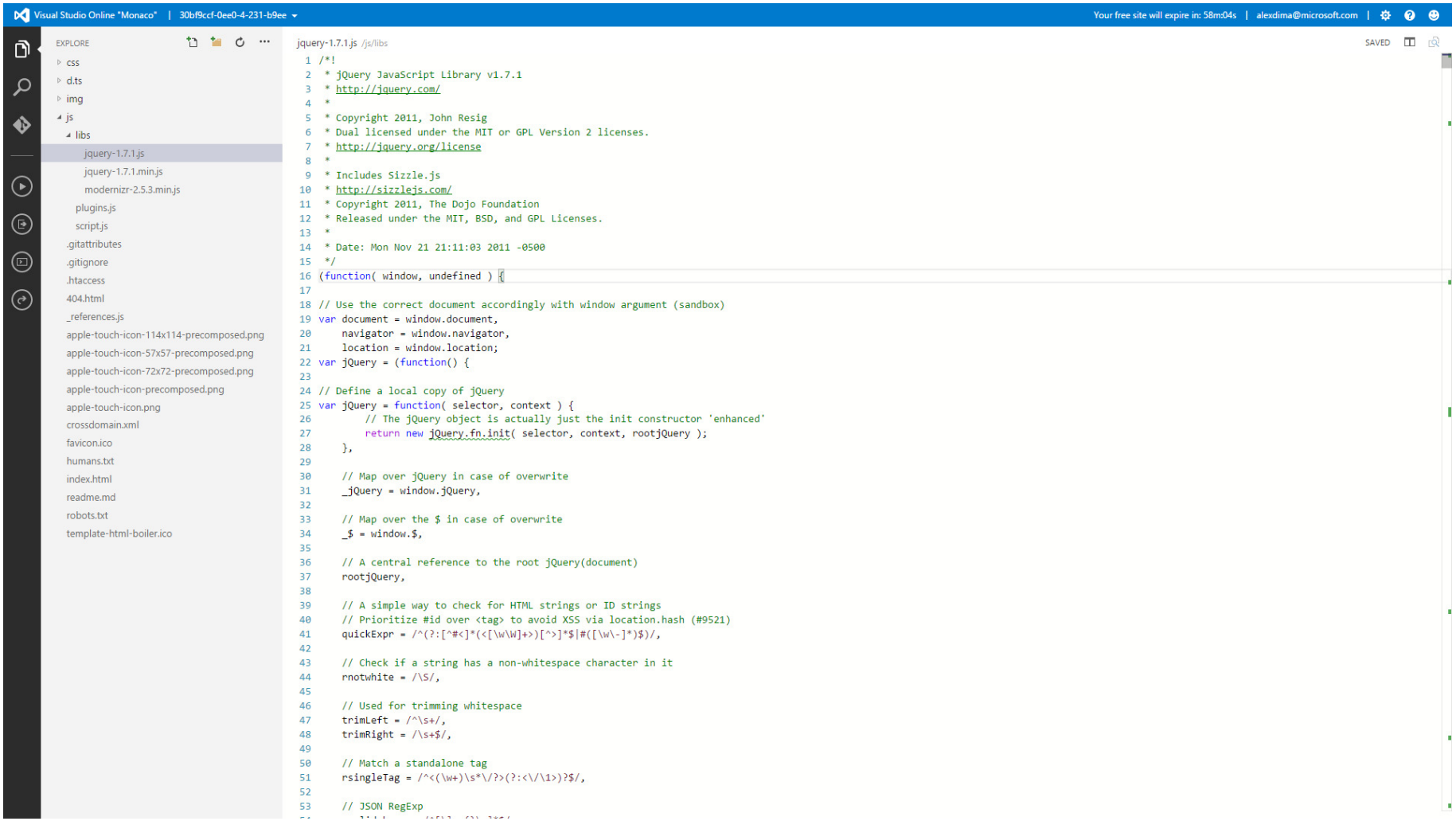
# Microsoft Edge



OneDrive


tasks - OneDrive
task.contribution.ts - Or
onedrive.live.com/?v=TextFileEditor&id=B66B7E86BEC6C0C%21395406&cid=0B66B7E86BEC6C0C&parId=B66B7E86BEC6C0C%21395403
OneDrive
Dirk Baeumer
Share
Download
Delete
Rename
Embed
Files > tasks > task.contribution
297 element.title = nls.localize('problems', "Problems");
298
299 Dom.addClass(error, 'task-statusbar-item-label-error');
300 error.innerHTML = '0';
301 label.appendChild(error);
302
303 Dom.addClass(warning, 'task-statusbar-item-label-warning');
304 warning.innerHTML = '0';
305 label.appendChild(warning);
306
307 Dom.addClass(info, 'task-statusbar-item-label-info');
308 label.appendChild(info);
309 $(info).hide();
310
311 callOnDispose.push(Dom.addDisposableListener(label, 'click', (e: MouseEvent) => {
312 const panel = this.panelService.getActivePanel();
313 if (panel && panel.getId() === Constants.MARKERS_PANEL_ID) {
314 this.partService.setPanelHidden(true);
315 } else {
316 this.panelService.openPanel(Constants.MARKERS_PANEL_ID, true);
317 }
318 }));
319
320 let updateStatus = (element: HTMLDivElement, stats: number): boolean => {
321 if (stats > 0) {
322 element.innerHTML = stats.toString();
323 $(element).show();
324 return true;
325 } else {
326 $(element).hide();
327 return false;
328 }
329 };
330
331
332 let manyMarkers = nls.localize('manyMarkers', "99+");
333 let updateLabel = (stats: MarkerStatistics) => {
334 error.innerHTML = stats.errors < 100 ? stats.errors.toString() : manyMarkers;
335 warning.innerHTML = stats.warnings < 100 ? stats.warnings.toString() : manyMarkers;
336 updateStatus(info, stats.infos);
337 };
338
339 this.markerService.onMarkerChanged((changedResources) => {
340 updateLabel(this.markerService.getStatistics());

# Visual Studio Online "Monaco"

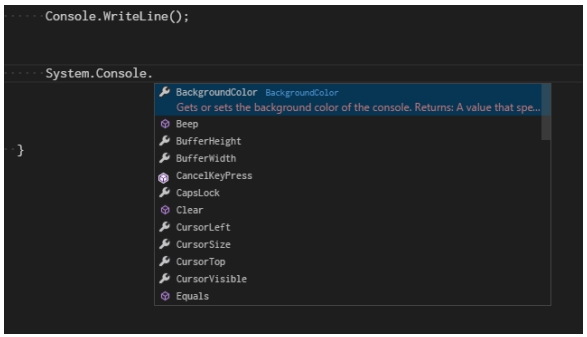


# Inside Visual Studio Code

- Electron


- TypeScript


- Monaco Editor



# ELECTRON

## Build cross platform desktop apps with web technologies

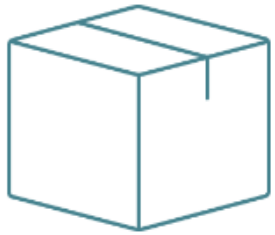
Formerly known as Atom Shell. Made with ♥ by GitHub.

### Web Tech

Use HTML, CSS, and JavaScript with Chromium and Node.js to build your app.

### Open Source

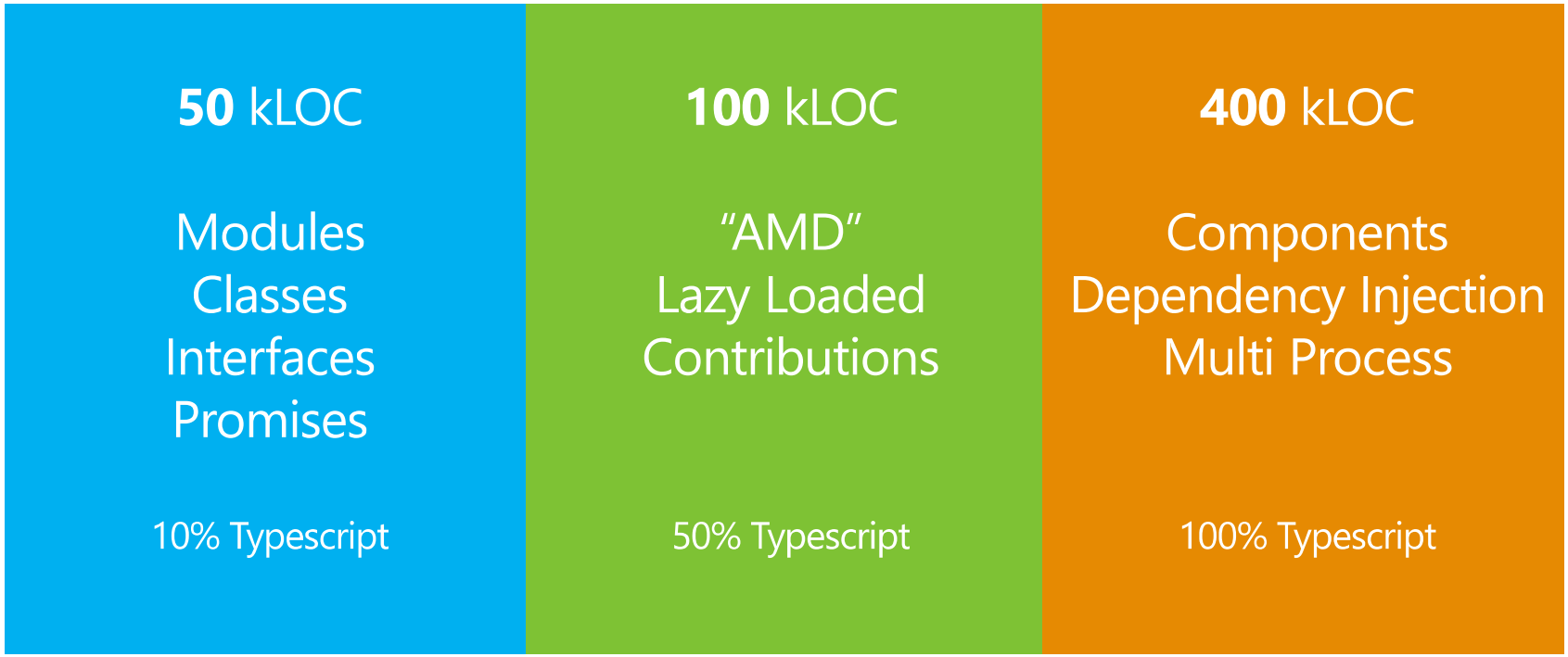
Electron is open source; maintained by GitHub and an active community.



### Cross Platform

Electron apps build and run on Mac, Windows, and Linux.

# Growing the Code




autumn
2011

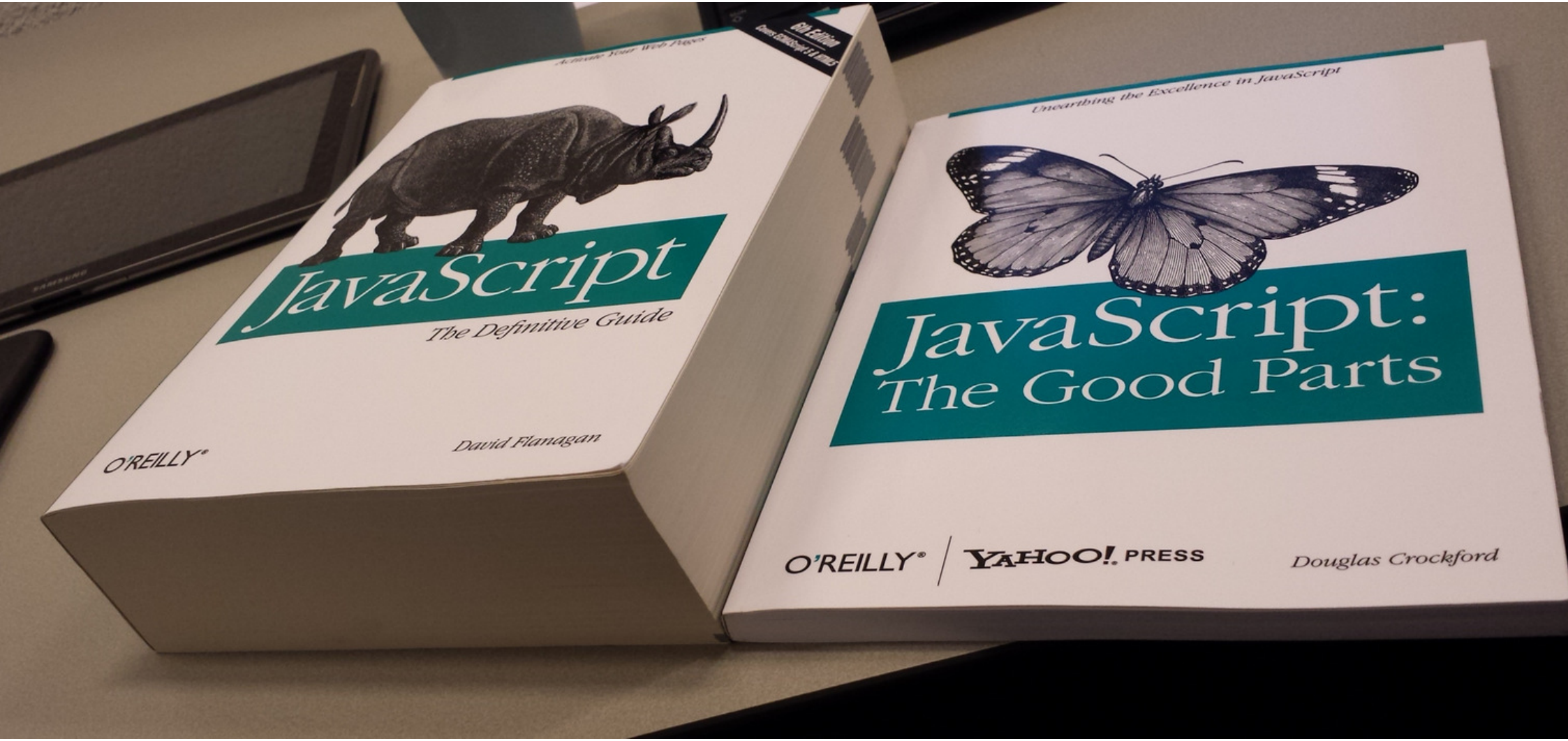
today
~800 LOC

# Growing the Code

**50** kLOC

Modules
Classes
Interfaces
Promises

10% Typescript


JavaScript
The Definitive Guide
David Flanagan
O'REILLY
JavaScript:
The Good Parts
Douglas Crockford
O'REILLY
YAHOO! PRESS

# Large scale JavaScript development is hard.

https://www.destroyallsoftware.com/talks/wat

# JavaScript Pains

Programming JavaScript is like carving code in stone. Refactoring it is difficult. JavaScript code rots over time.

Comprehending JavaScript code is difficult as well (hard to browse and navigate)

Large code bases need to come up with compensating patterns for classes (es6), modules (amd or commonjs) and namespaces.

Describing APIs means keeping documentation synchronized with the implementation.

## Starts with JavaScript

- All JavaScript code is TypeScript code, simply copy and paste
- All JavaScript libraries work with TypeScript

## Optional static types, classes, modules

- Enable scalable application development and excellent tooling
- Zero cost: Static types completely disappear at run

# TypeScript to the Rescue…

Structural typing, type inference, optional types

Zero run time cost, types disappear at run time

```
1 reference
1 interface Point {
      0 references
2 →   x: number;
      0 references
3 →   y: number;
4 }
5
6 let p = { x: 100, y: 100, description: "..." };
7 let point: Point = p;
```

```
 9 let person = {
10 →   cyclist: 'Johannes',
11 →   bikes: [
12 →   →   'Santa Cruz Nomad CC',
13 →   →   'Intense Carbine 29'
14 →   ]
15 };
16
17 let bikes = person.bikes;
18 let rider = person.rider;
```

(property) bikes: string[]

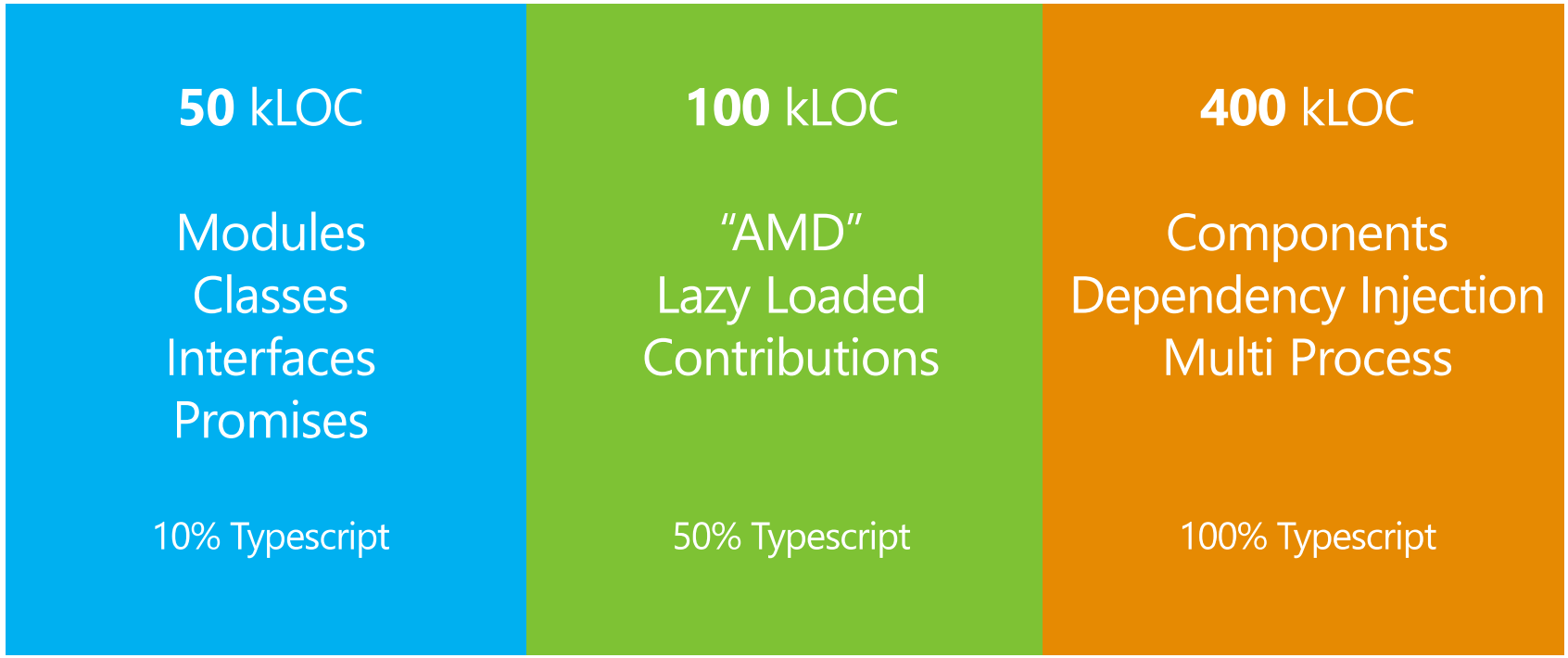
(1/1) [ts] Property 'rider' does not exist on type '{ cyclist: string; bikes: string[]; }'.

# Makes ES6 features available in ES3/5 runtimes

- ✓ Arrows
- ✓ Classes
- ✓ Modules (import / export)
- ✓ Let + Const
- ✓ For..Of
- ✓ Destructuring
- ✓ Template Strings
- ✓ Default + Rest

See also: https://github.com/lukehoban/es6features

# Growing the Code




autumn
2011

today

# Growing the Code

**100** kLOC

"AMD"
Lazy Loaded
Contributions

50% Typescript

# Code Organization: Keep code structured

```
1 var Namespace = {};
2 Namespace.Util = {};
3 Namespace.Util.Strings = {};
4 Namespace.Util.Strings.trim = function() {
5   /* etc */
6 }
```

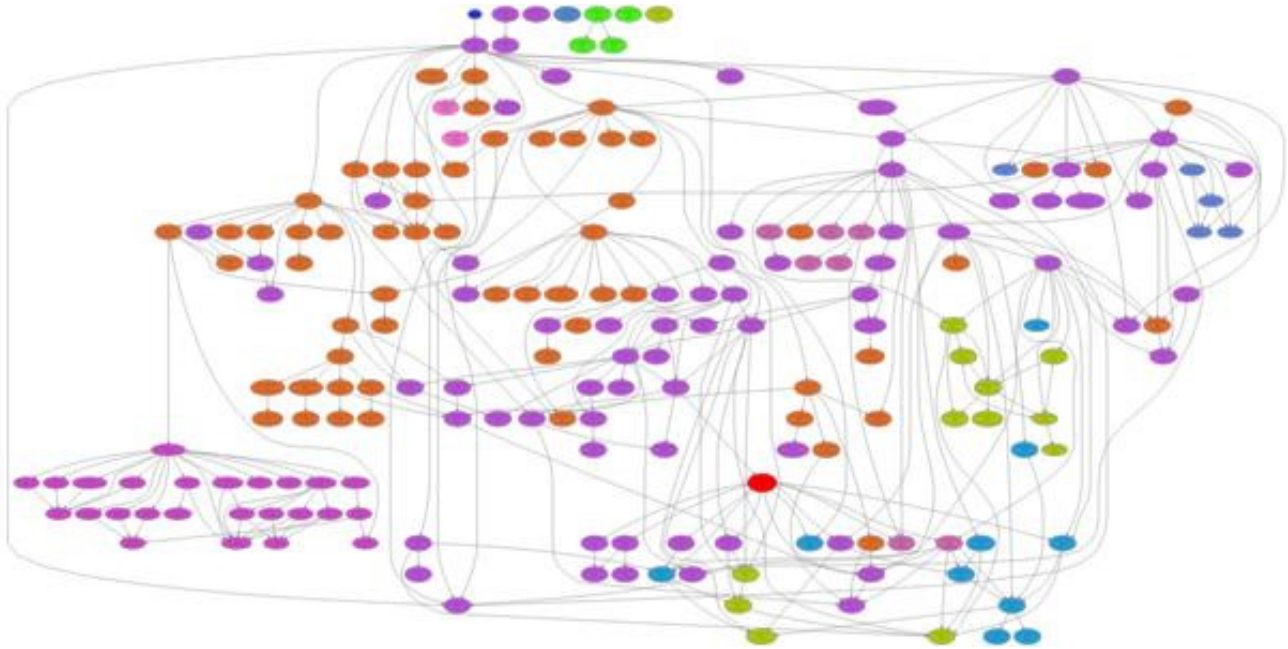
Global namespaces are a pain

No relationship to the source files on disk

Renaming things was a pain

Cycling dependencies were easily unnoticed

# Code Organization: Dependencies



*Our dependency graph was such a mess that each area had a dependency on just about every other area*

# Module Systems to the Rescue...

## AMD

```
1 define('module_id', [
2     'dependency_id'
3 ], function(dependency) {
4     // code
5
6     return {
7         // exports
8     };
9 });
```

## CommonJS

my_module.js

```
1 var dependency = require('dependency_id');
2
3 // code
4
5 exports.myExports = {
6     // exports
7 }
```

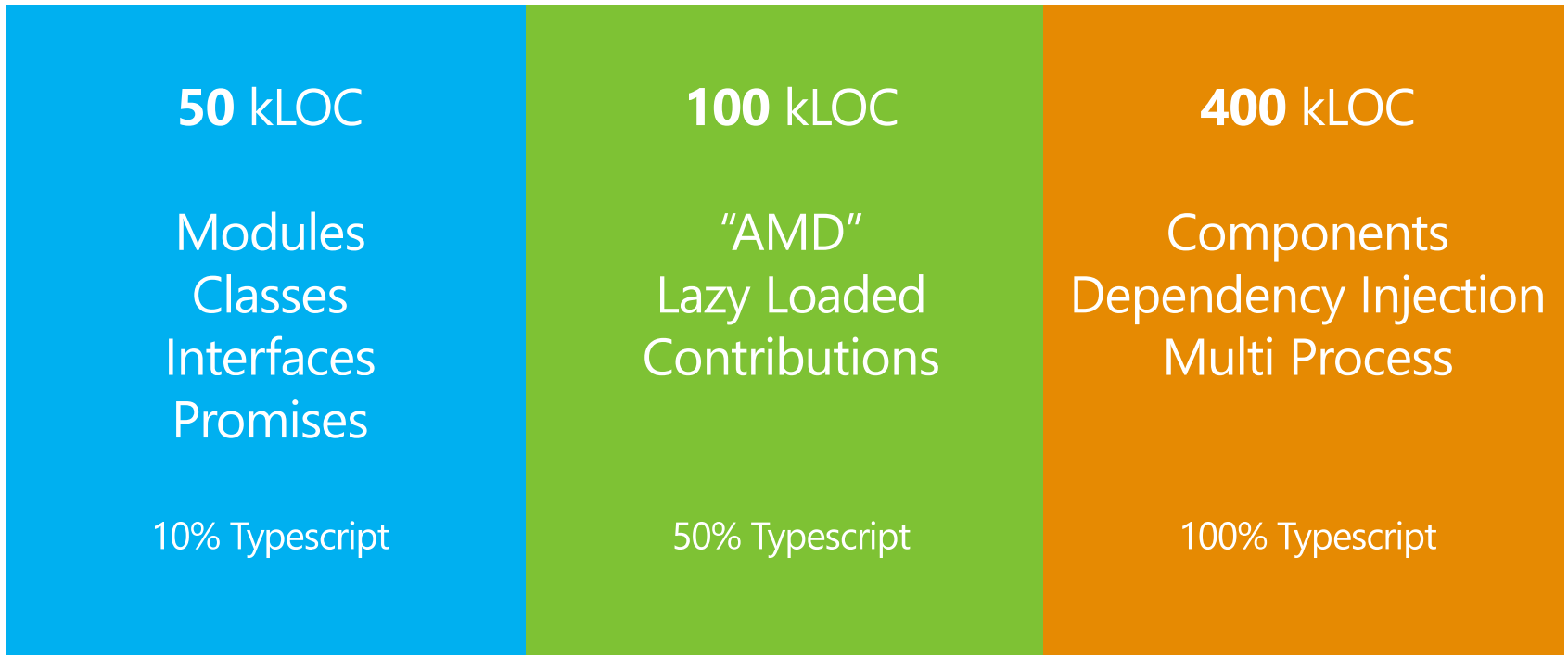
# TypeScript: First class module support

Supports both AMD and CommonJS with one syntax

```
1 import { Dependency } from "./dependency_id";
2
3 // code
4
5 export function helper() {
6     // export
7 }
```

Sharing code between AMD and CommonJS is easy

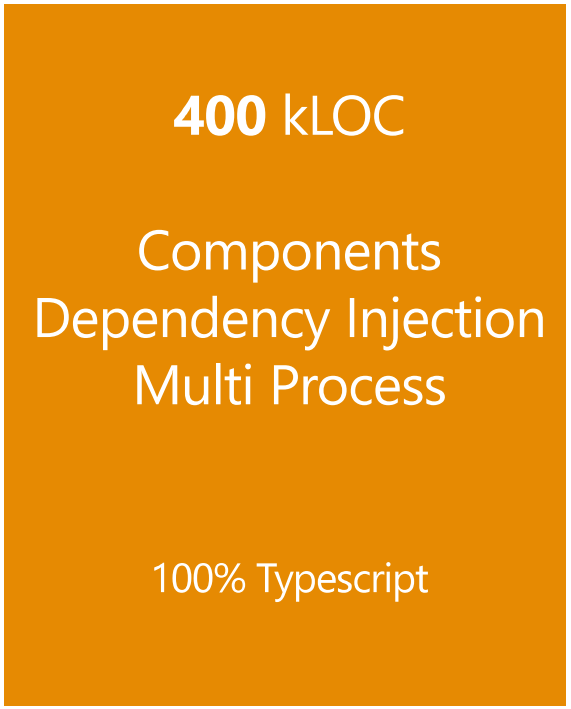
# Growing the Code




autumn
2011

today

# Growing the Code




today

# 100% TypeScript

Incremental conversion from JavaScript

Establish & enforce rules: no implicit any

Conversion is also code clean up and fixing

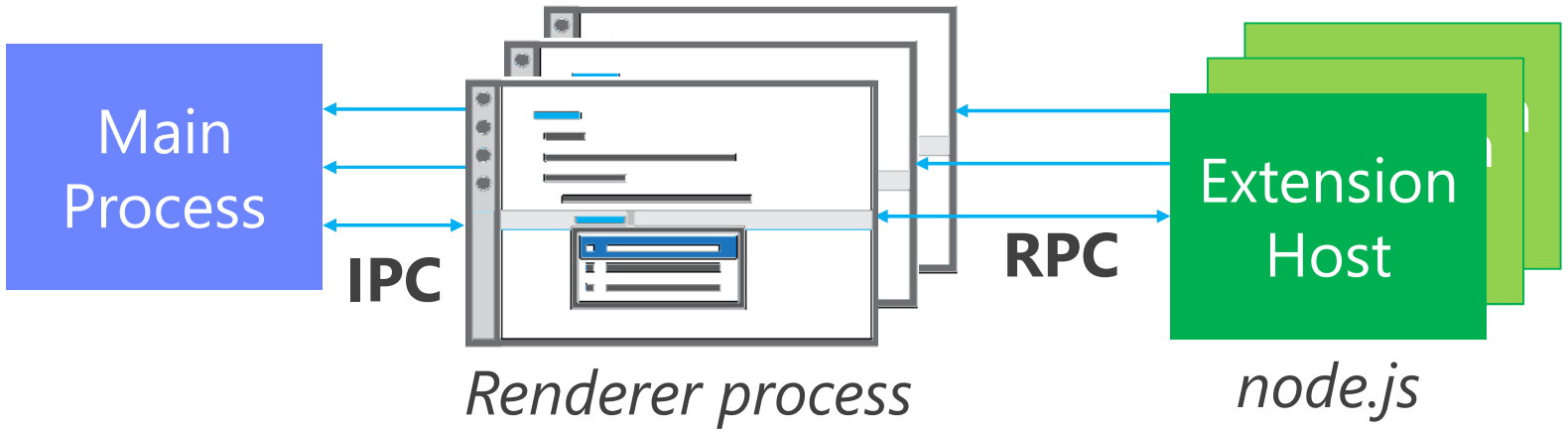
```
2 delete this.markers[range.statMarkerId];
3 // or is it startMarkerId?
```

# Dependency Injection

## Constructor service injection in many places

```
1773 export class ShowActiveFileInExplorer extends Action {
1774
1775     constructor(
1776         id: string,
1777         label: string,
1778         @IWorkbenchEditorService private editorService: IWorkbenchEditorService,
1779         @IViewletService private viewletService: IViewletService,
1780         @IWorkspaceContextService private contextService: IWorkspaceContextService,
1781         @IMessageService private messageService: IMessageService
1782     ) {
1783         super(id, label);
1784     }
```

# Extension Isolation




Controlled extensibility

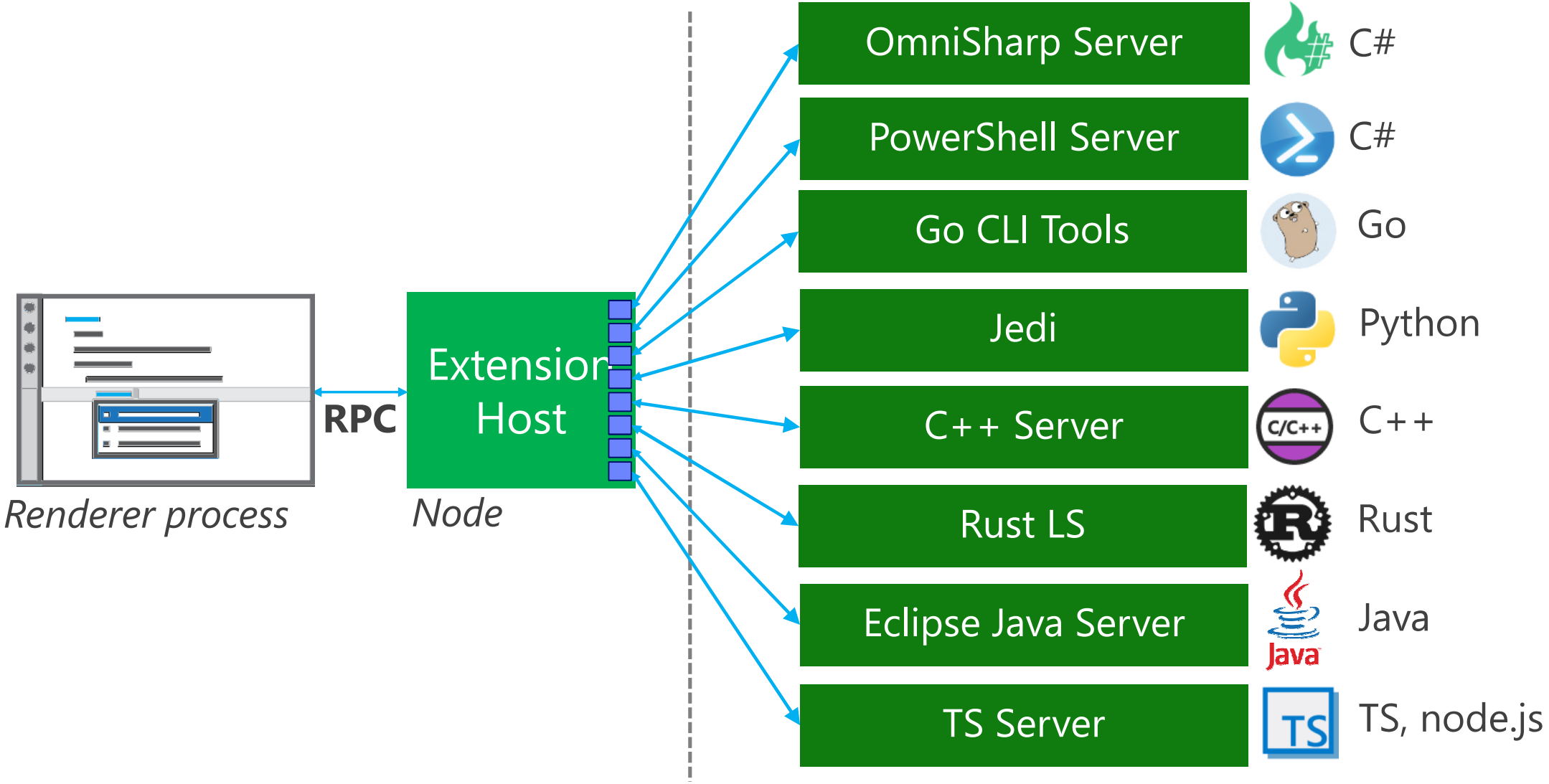
JavaScript or TypeScript

Reuse node modules

API: **vscode.d.ts**

Extension description: **package.json**

# Language Extensions: Language Servers



# Thanks!

@code

http://code.visualstudio.com

https://github.com/Microsoft/vscode/

http://www.typescriptlang.org/