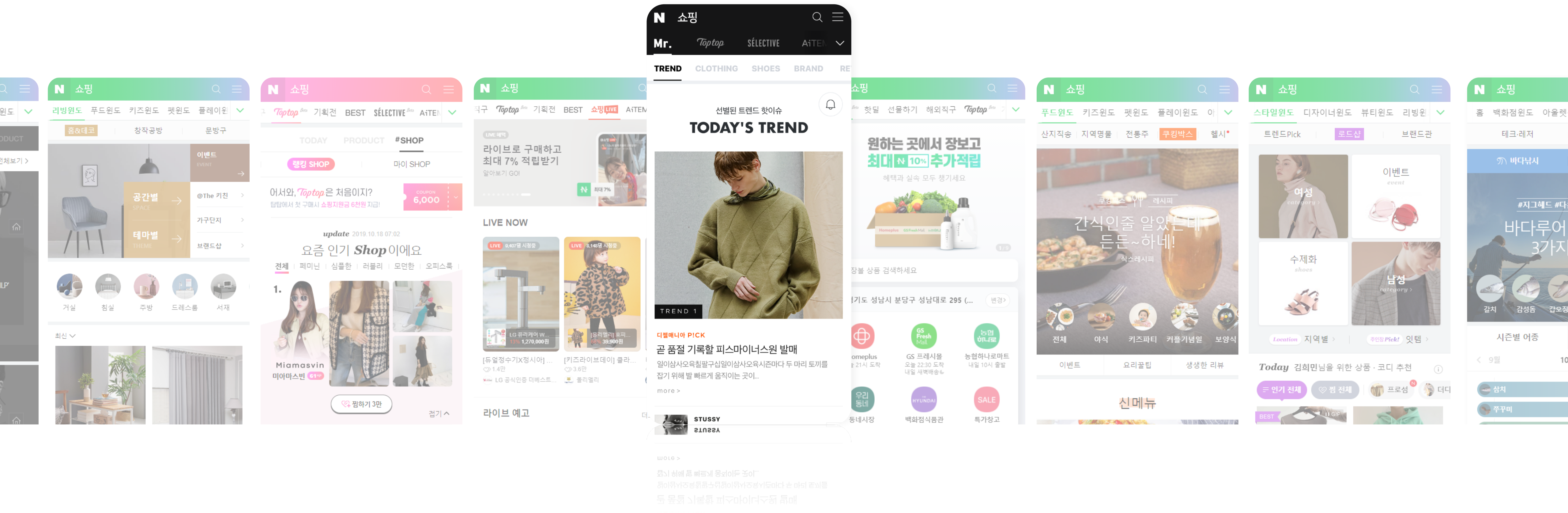


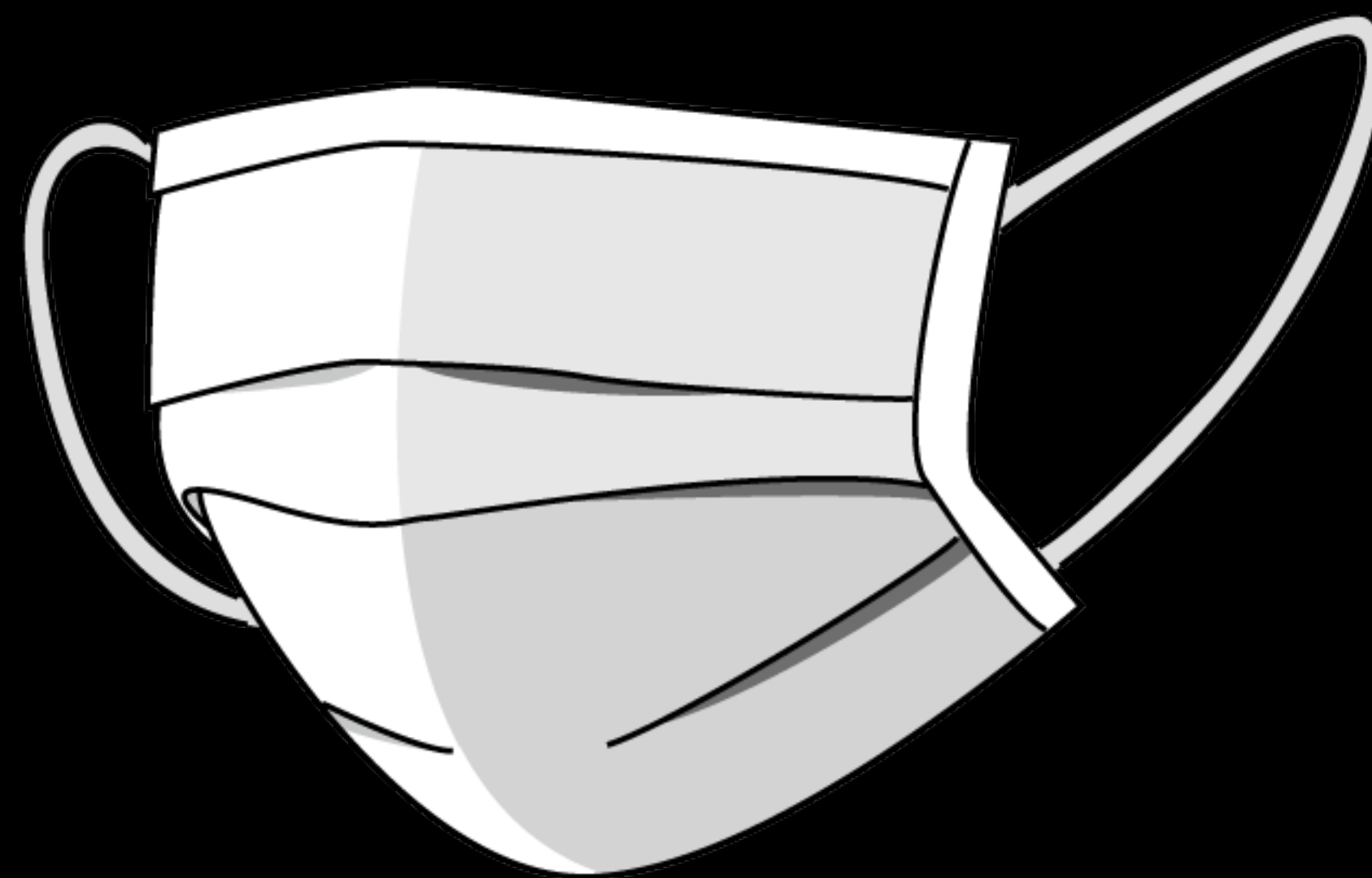
Deno를 통해 알아보는 JavaScript 세상 이야기

CONTENTS

1. What is Deno.
2. Javascript, Typescript
3. Node.js
4. About Deno
5. Coding :: Example
6. Preparing for a future with _____.

네이버 쇼핑 상품수 1억 3천만 스토어 60만 방문자 일 1억명 PV



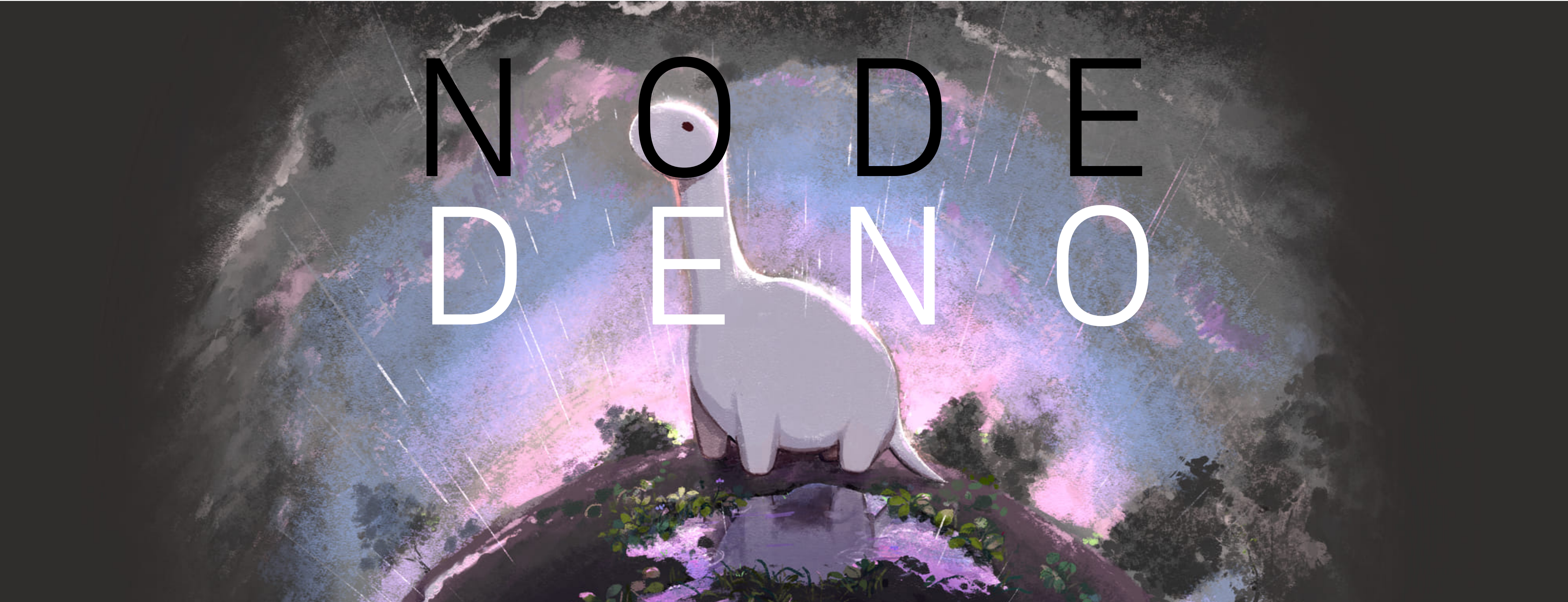


Our life has changed



Today I Speak

N DEVIEW
2020

A white dinosaur, possibly a Triceratops, stands on a small, grassy hill. The sky is dark and stormy, with heavy rain falling. The sky is illuminated with vibrant purple and blue light, creating a dramatic atmosphere. The dinosaur is positioned in the center of the frame, facing forward. The text "N O D E" is written in large, black, sans-serif capital letters across the top of the dinosaur's head and neck. The text "D E N O" is written in large, white, sans-serif capital letters across the middle of the dinosaur's body. The overall scene is a surreal, artistic representation of a dinosaur in a modern, digital environment.

N O D E
D E N O

1. What is Deno

Deno

N DEVIEW
2020



A secure runtime for javascript and typescript

Js/Ts

V8

Rust

2. JavaScript, TypeScript



JS

History



1994년

1995년

JavaScript

JScript



History



1994년

1995년

JavaScript

JScript



History



1994년

1995년

2004년

2008년

JavaScript

JScript

ActionScript

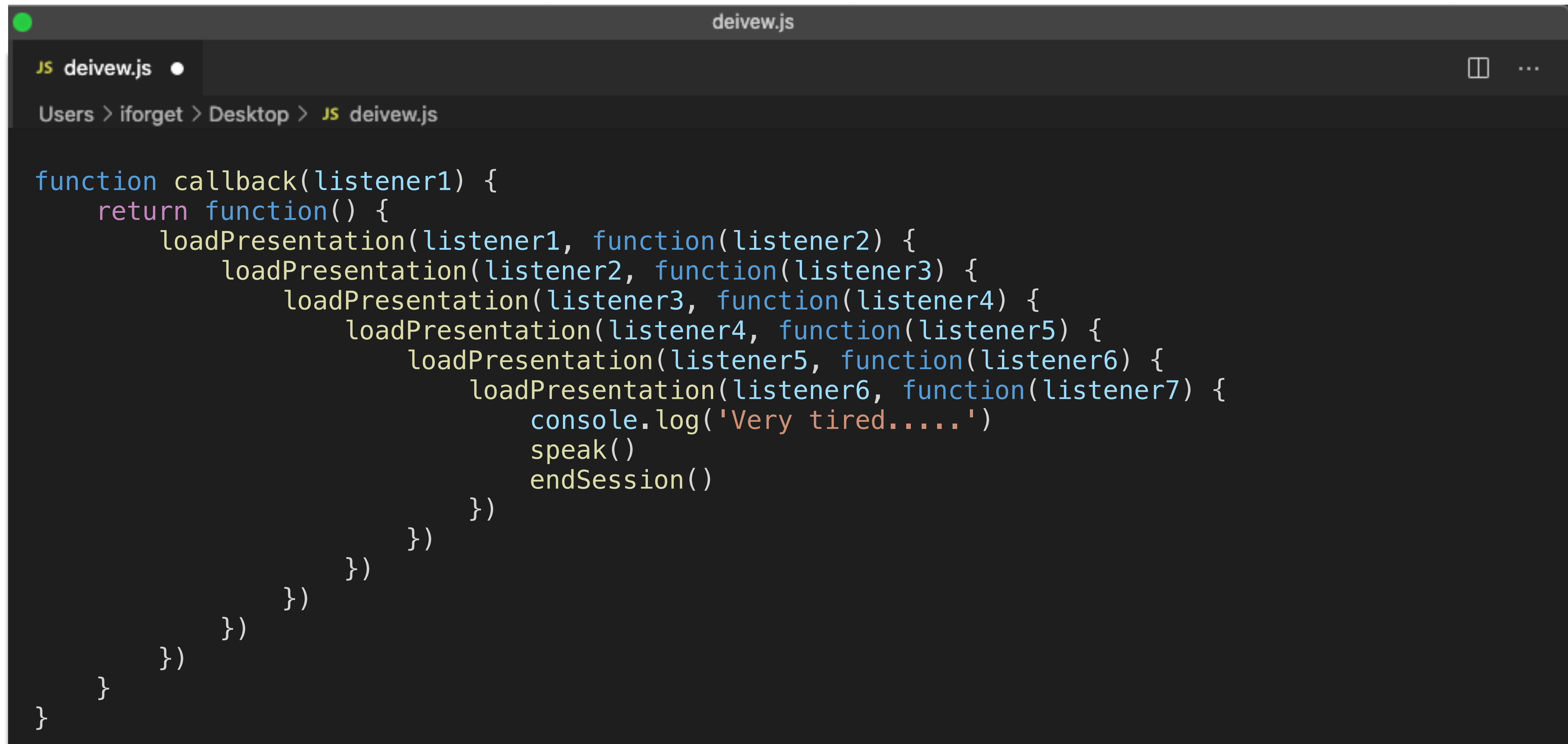


ECMAScript 2009 (ES5)



[소스] pngegg.com image

Inconvenient



The screenshot shows a code editor window titled 'deivew.js'. The breadcrumb path is 'Users > iforget > Desktop > JS deivew.js'. The code is a JavaScript function named 'callback' that takes 'listener1' as an argument. It returns an anonymous function that calls 'loadPresentation' with 'listener1' and a nested function. This pattern repeats for seven levels of nesting, with the final innermost function calling 'console.log('Very tired.....')', 'speak()', and 'endSession()'. The closing braces for all functions are aligned to the left, creating a staircase effect that illustrates the 'callback hell' problem.

```
function callback(listener1) {  
  return function() {  
    loadPresentation(listener1, function(listener2) {  
      loadPresentation(listener2, function(listener3) {  
        loadPresentation(listener3, function(listener4) {  
          loadPresentation(listener4, function(listener5) {  
            loadPresentation(listener5, function(listener6) {  
              loadPresentation(listener6, function(listener7) {  
                console.log('Very tired.....')  
                speak()  
                endSession()  
              })  
            })  
          })  
        })  
      })  
    })  
  })  
}
```

Callback

Inconvenient

```
deivew.js
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

var deivew0bj = {
  node : '2008',
  deno : '2020',
  done : 'my situation',
  endo : 'our situation',
}

var done = deivew0bj.done
var endo = deivew0bj.endo
var myResult = {
  dent: deivew0bj.deno,
  done: deivew0bj.done
}
```

~~Handmade~~
Destructuring assignment

```
deivew.js
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

var name = 'KeunBae bae'
var age = 27
var email = 'keunbae.bae@navercorp.com'

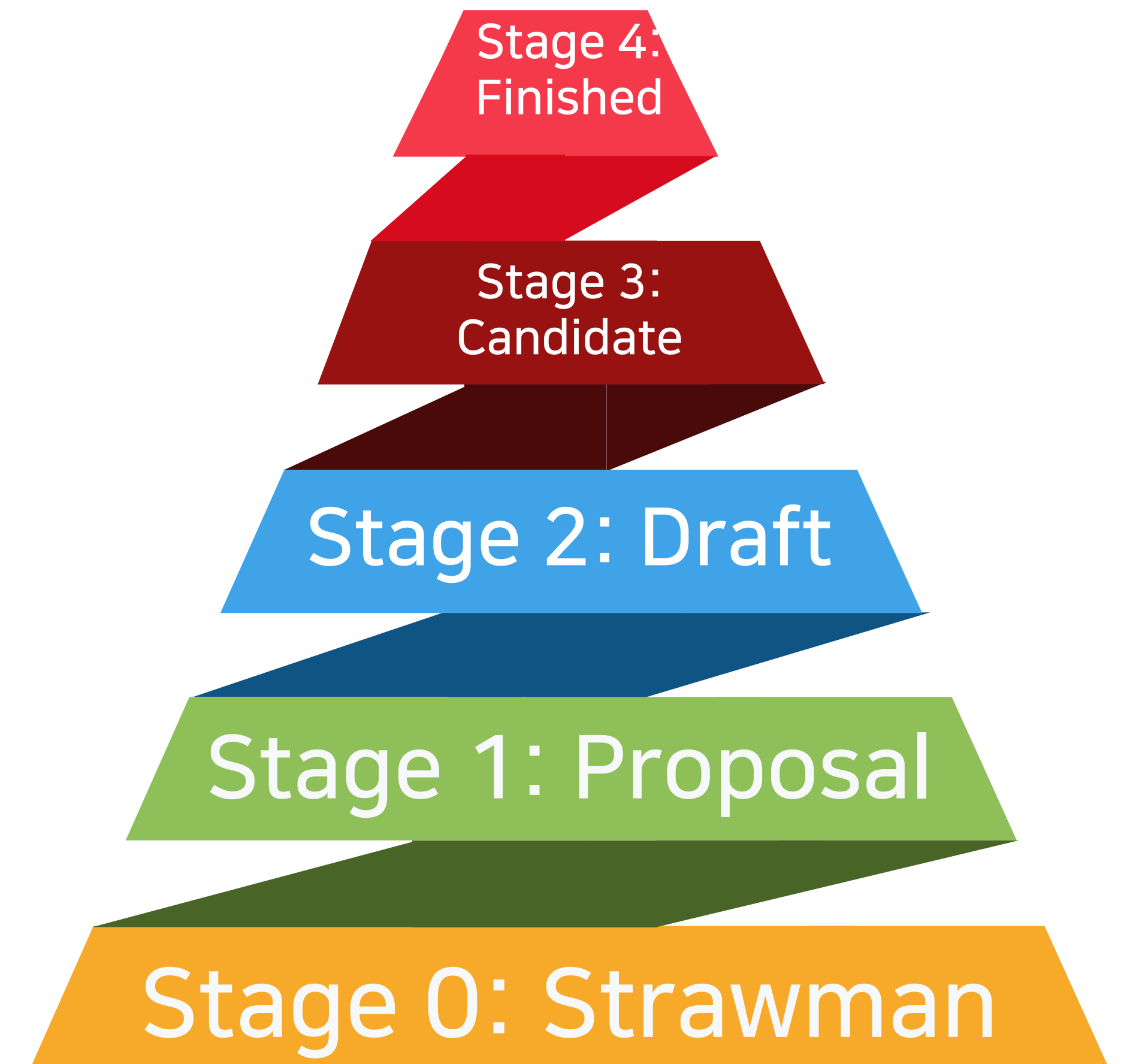
var datas = {
  name: name,
  age: age,
  email: email
}
```

Object Initialize

Improvement steps



Stage 0: Strawman
누구나 의견을 제시할 수 있음.

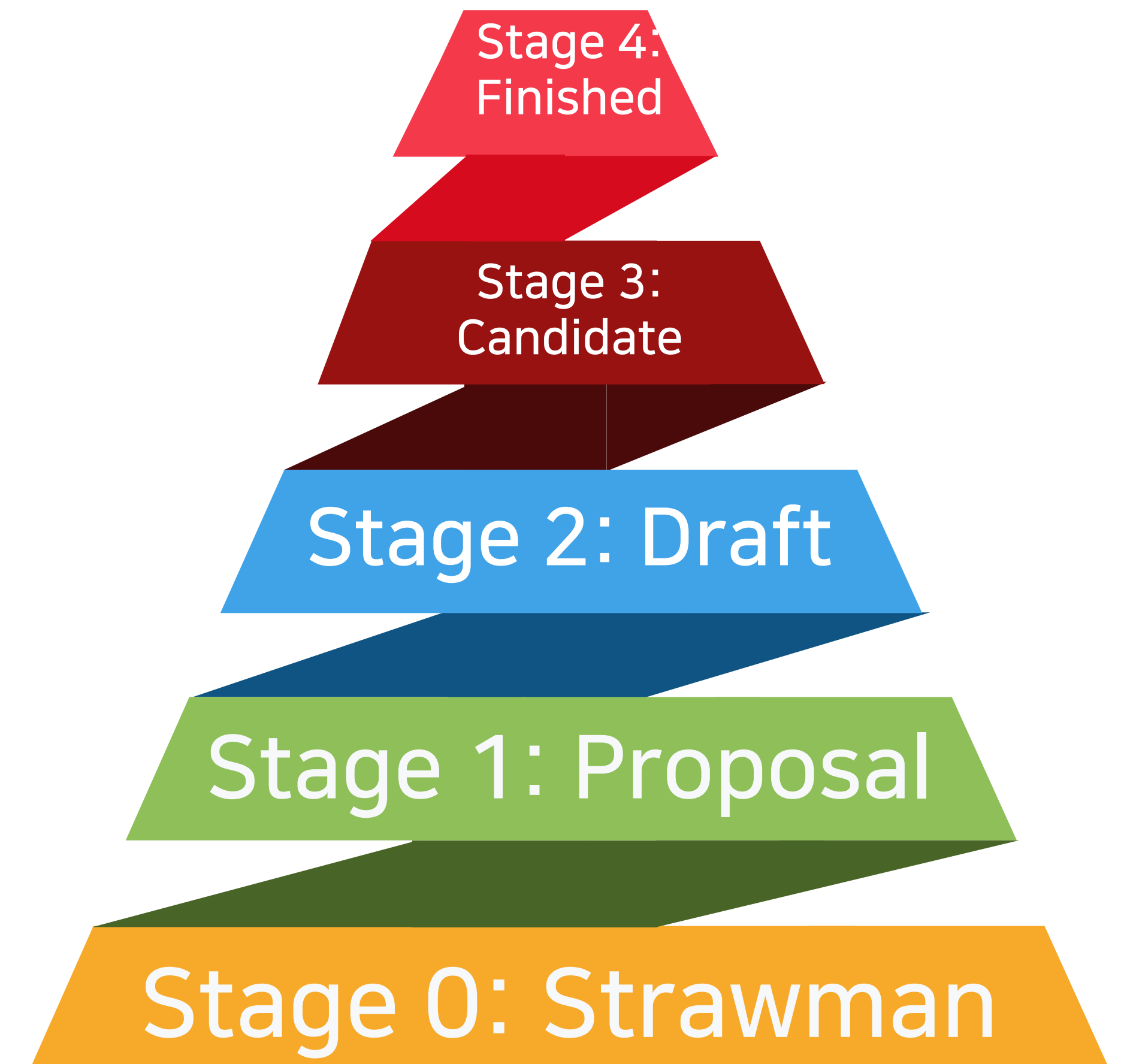


Improvement steps



Stage 1: Proposal

해결하고자 하는 문제와 하イレ벨
API 단계 제시

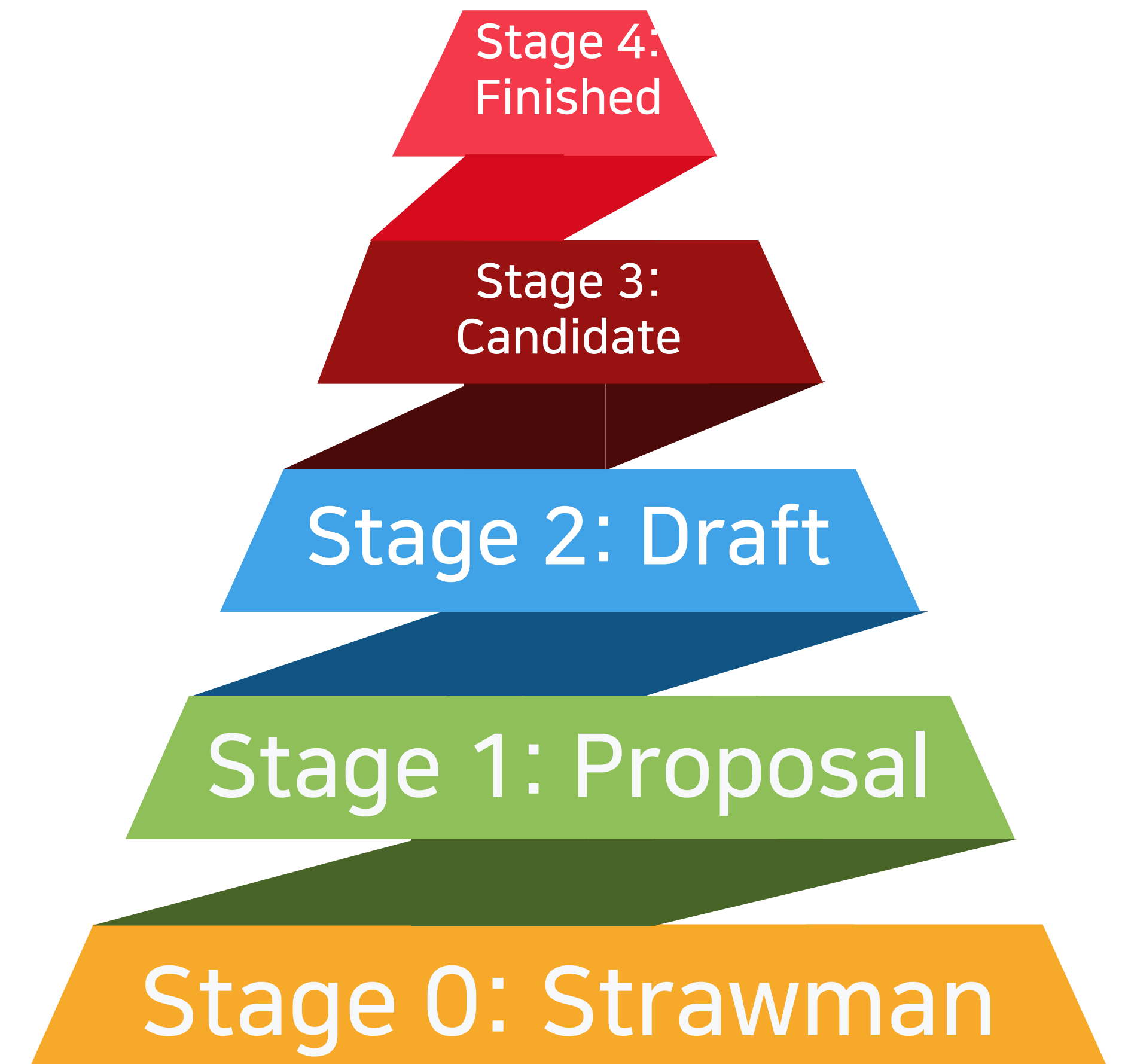


Improvement steps



Stage 2: Draft

ECMA의 표준의 언어로 작성된
형식적인 서술 초안 작성

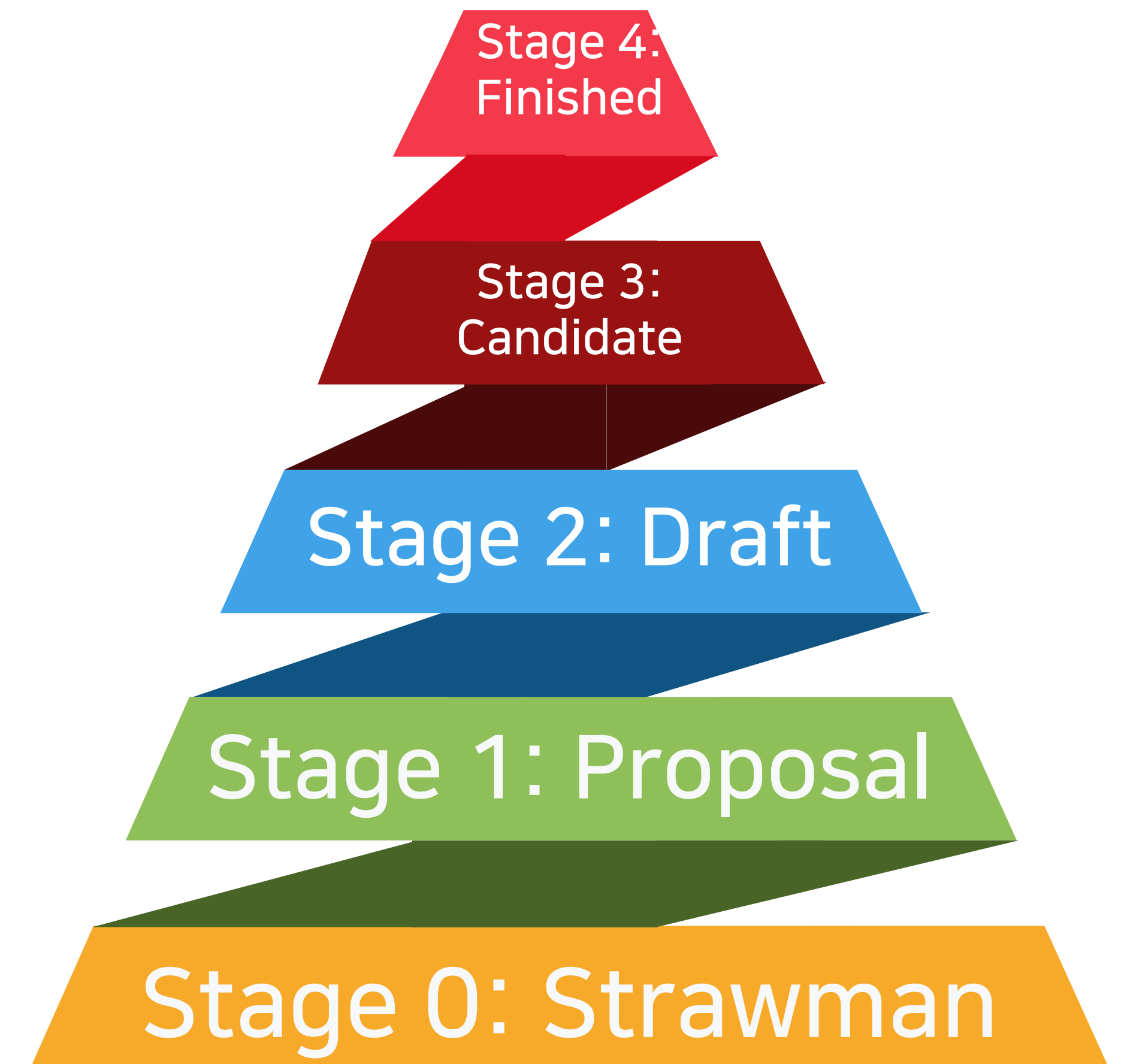


Improvement steps



Stage 3: Candidate

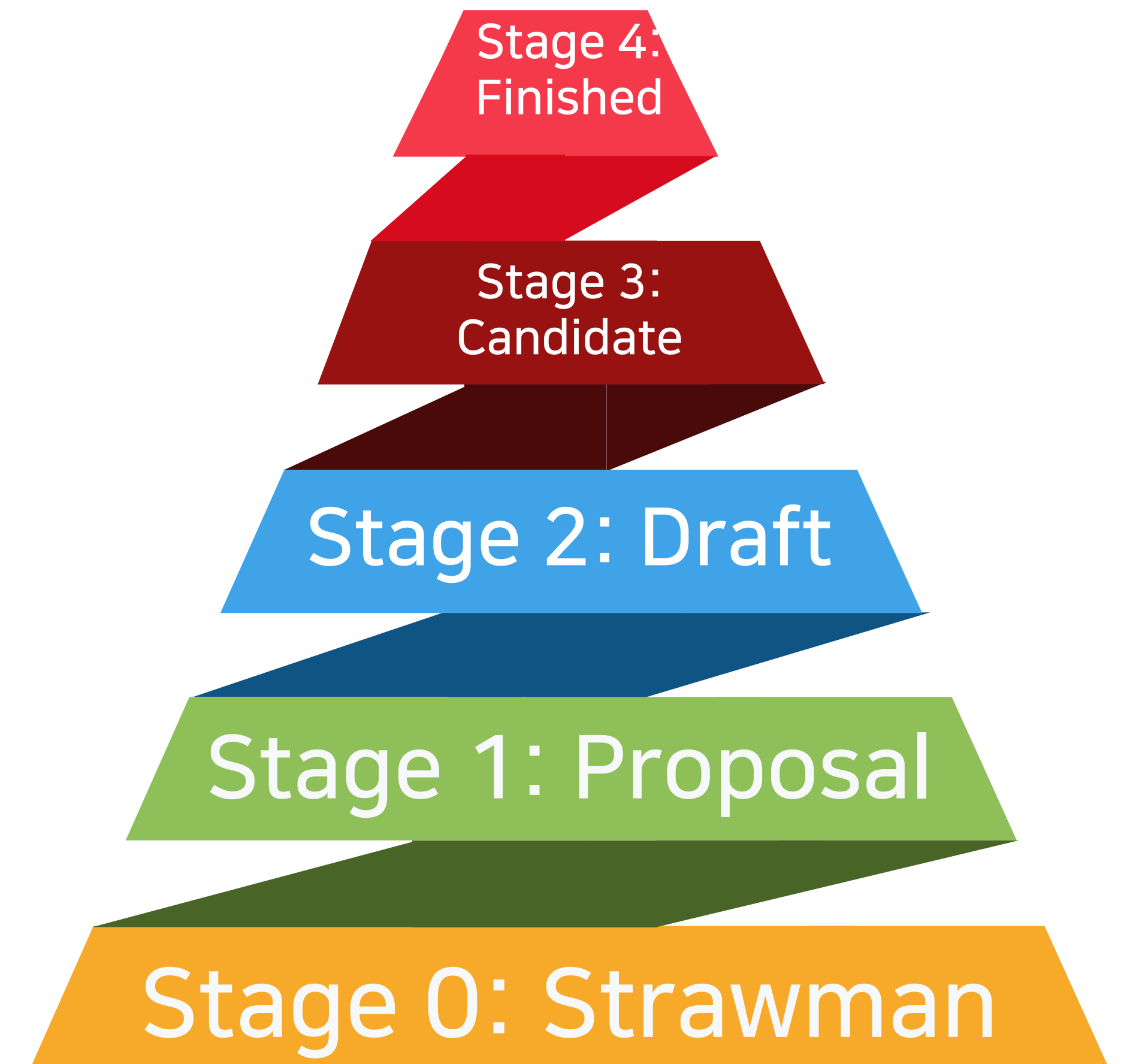
대부분 완성에 가깝고 피드백을
수정하며 마무리된 명세를 작성



Improvement steps



Stage 4: Finished
테스트가 작성되고 최소 2개 이상
의 상세구현 시 표준 정의반영



Resolution

```
deivew.js
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

async function asyncFunction(li1) {
  const li2 = await loadPresentation(li1)
  const li3 = await loadPresentation(li2)
  const li4 = await loadPresentation(li3)
  const li5 = await loadPresentation(li4)
  const li6 = await loadPresentation(li5)
  const li7 = await loadPresentation(li7)
  console.log('not tired.....')
  speak()
  endSession()
}
```

async/await

```
deivew.js
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

let devviewObj = {
  node : '2008',
  deno : '2020',
  done : 'my situation',
  endo : 'our situation',
}
const { node, deno, ...myResult } = devviewObj

let name = 'KeunBae bae'
let age = 27
let email = 'keunbae.bae@navercorp.com'

const datas = { name, age, email }
```

Destructuring assignment
Object Initialize

Limitations of JavaScript

동적 언어 특성상 ...

- 우리는 단지 "3" 이라는 결과를 원했을 뿐인데..

```
deivew.js
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

function add (val1, val2) {
    return val1 + val2
}

console.log(add(1, 2))
// result : 3

console.log(add("de", "view"))
// result : devview

console.log(add("1", 2))
// result : 12

console.log(add(1))
// result : NaN
```

Limitations of JavaScript



IDE

```
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

class Duck {
  quack() {
    console.log('꽹!');
  }
}

class Person {
  quack() {
    console.log('나는 꺽');
  }
}

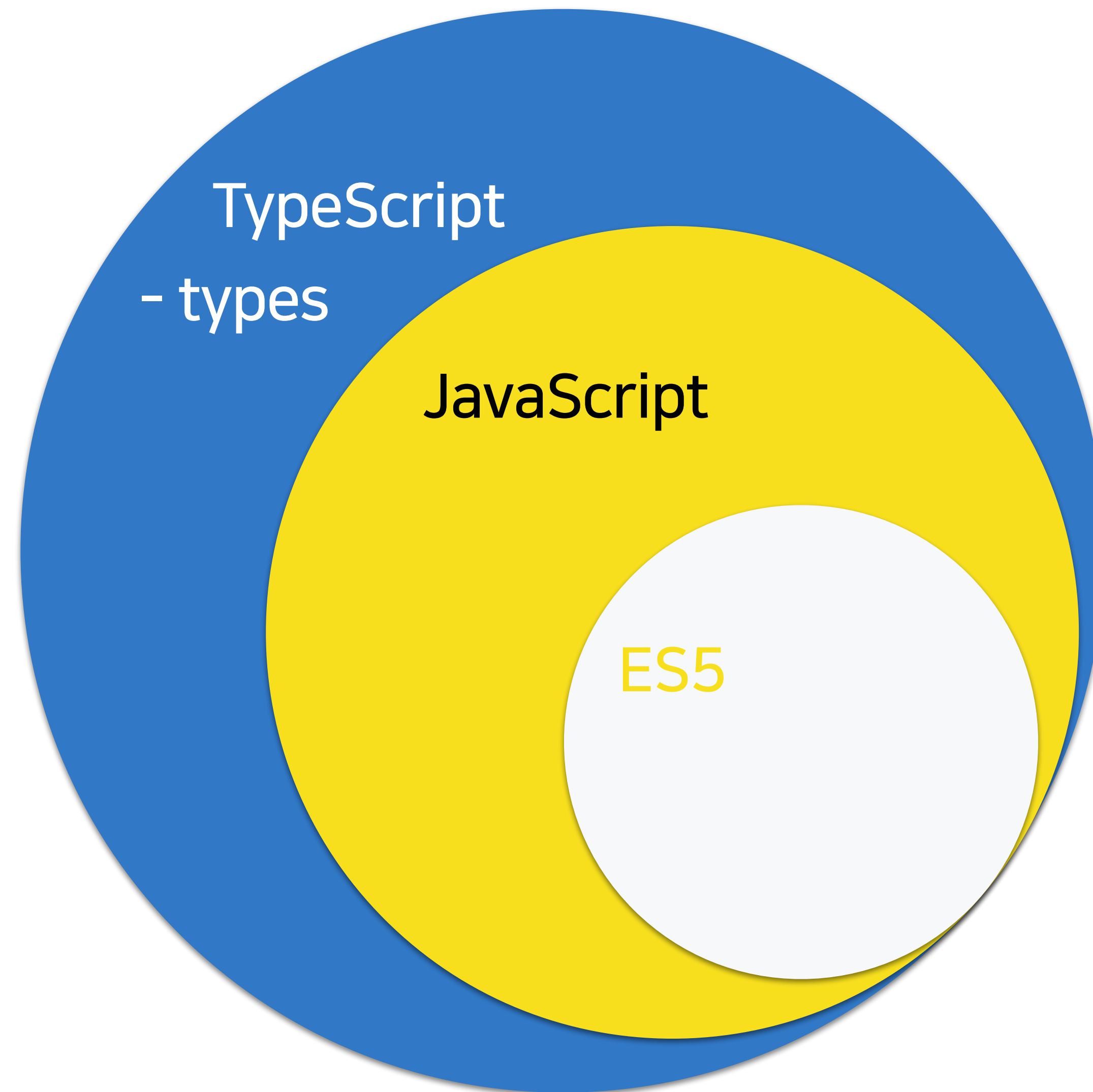
function makeSomeNoiseWith(duck) {
  duck.quack();
}

makeSomeNoiseWith(new Duck()); // OK
makeSomeNoiseWith(new Person()); // OK
```

Interface

T S

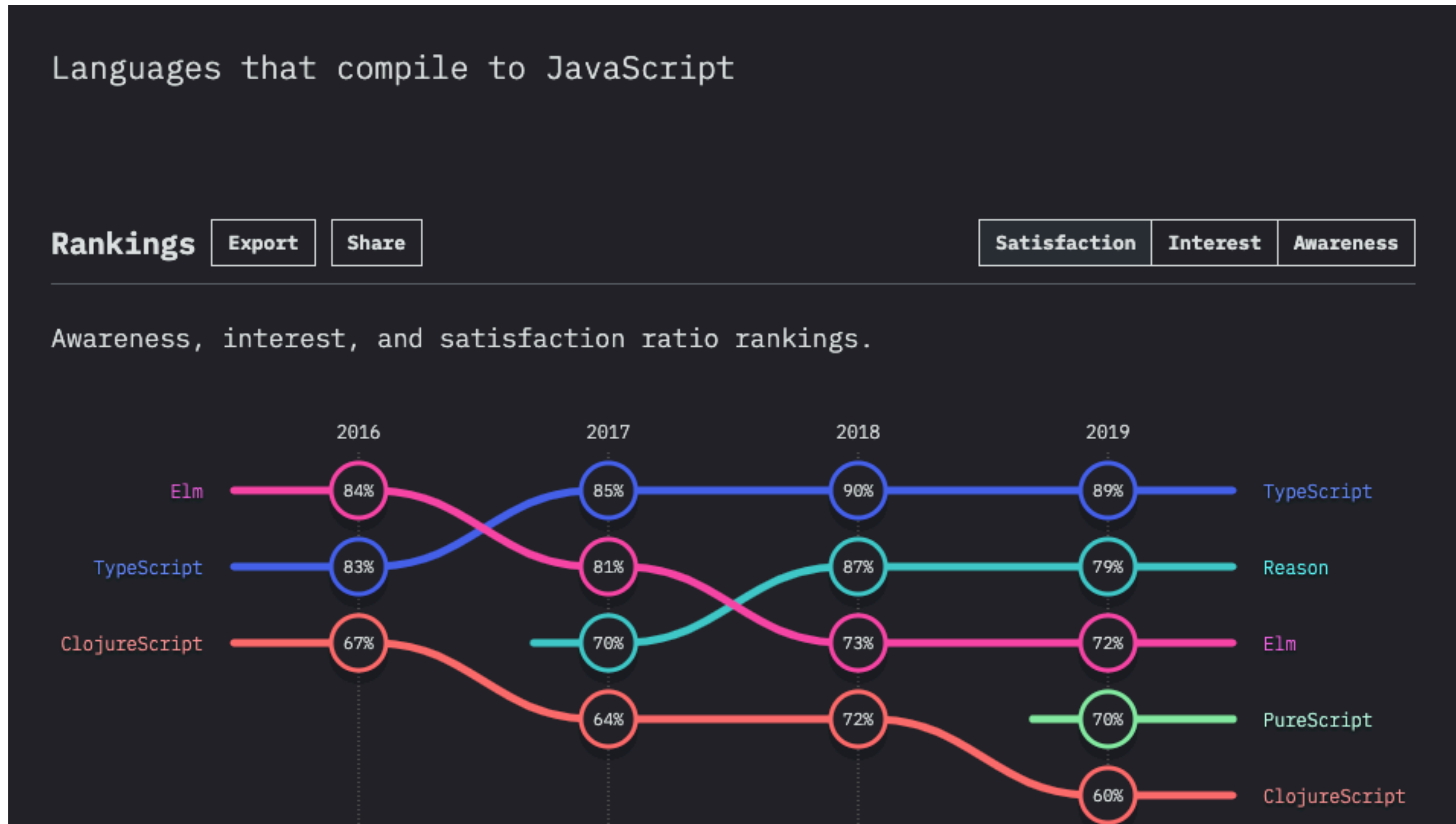
TypeScript



We like TypeScript



We like TypeScript

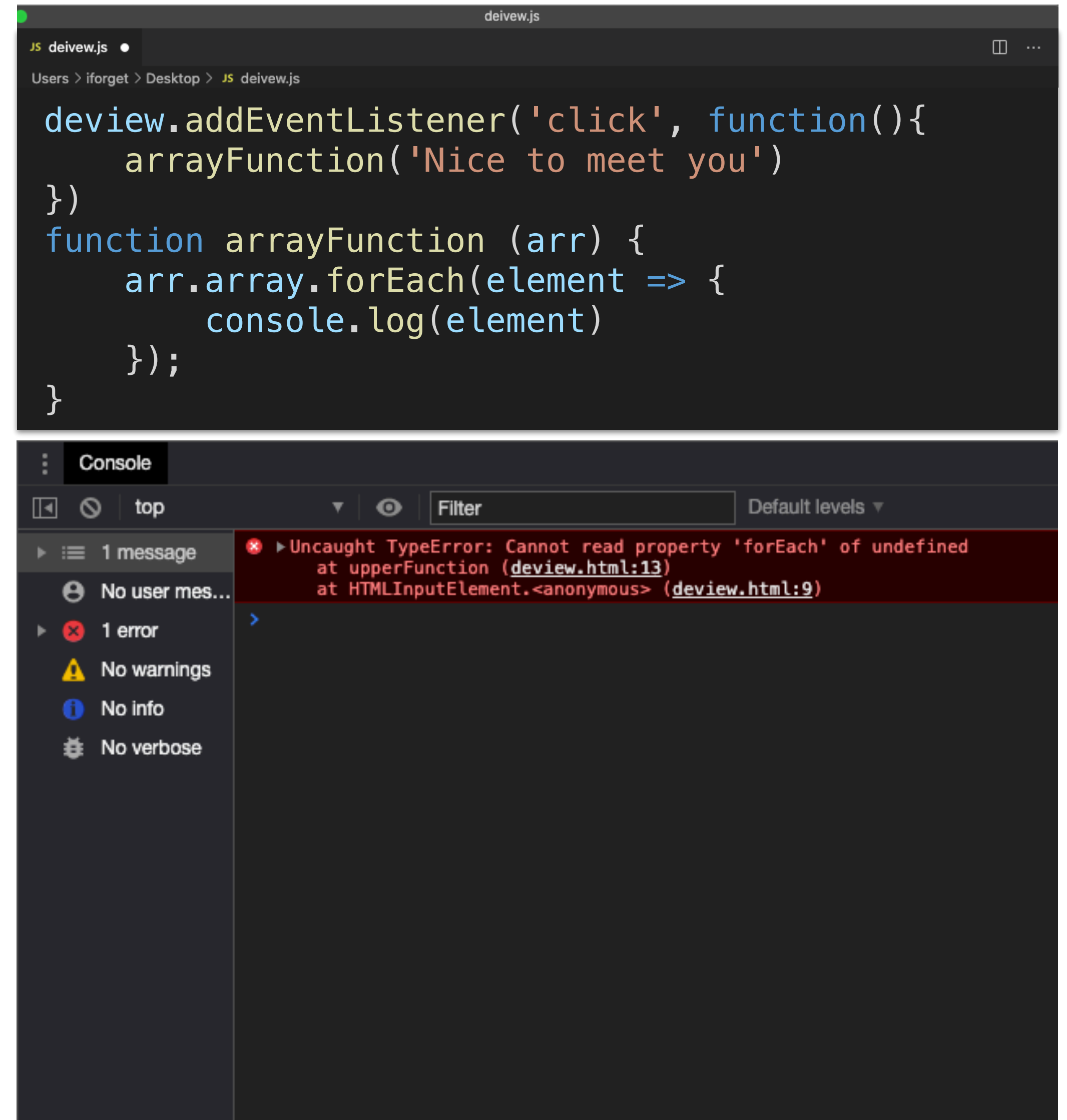


[참조] state of JS 2019 캡처

Need :: Human Error

HOXY 이런 에러들을 본 적 없으신
가요?

- 타입을 잘못 지정하거나



The screenshot shows a web browser window with a dark theme. The address bar shows the file path: Users > iforget > Desktop > JS deivew.js. The main content area displays the following JavaScript code:

```
deview.addEventListener('click', function(){
    arrayFunction('Nice to meet you')
})
function arrayFunction (arr) {
    arr.array.forEach(element => {
        console.log(element)
    });
}
```

The browser's developer console is open, showing a red error message:

```
Uncaught TypeError: Cannot read property 'forEach' of undefined
    at upperFunction (deview.html:13)
    at HTMLInputElement.<anonymous> (deview.html:9)
```

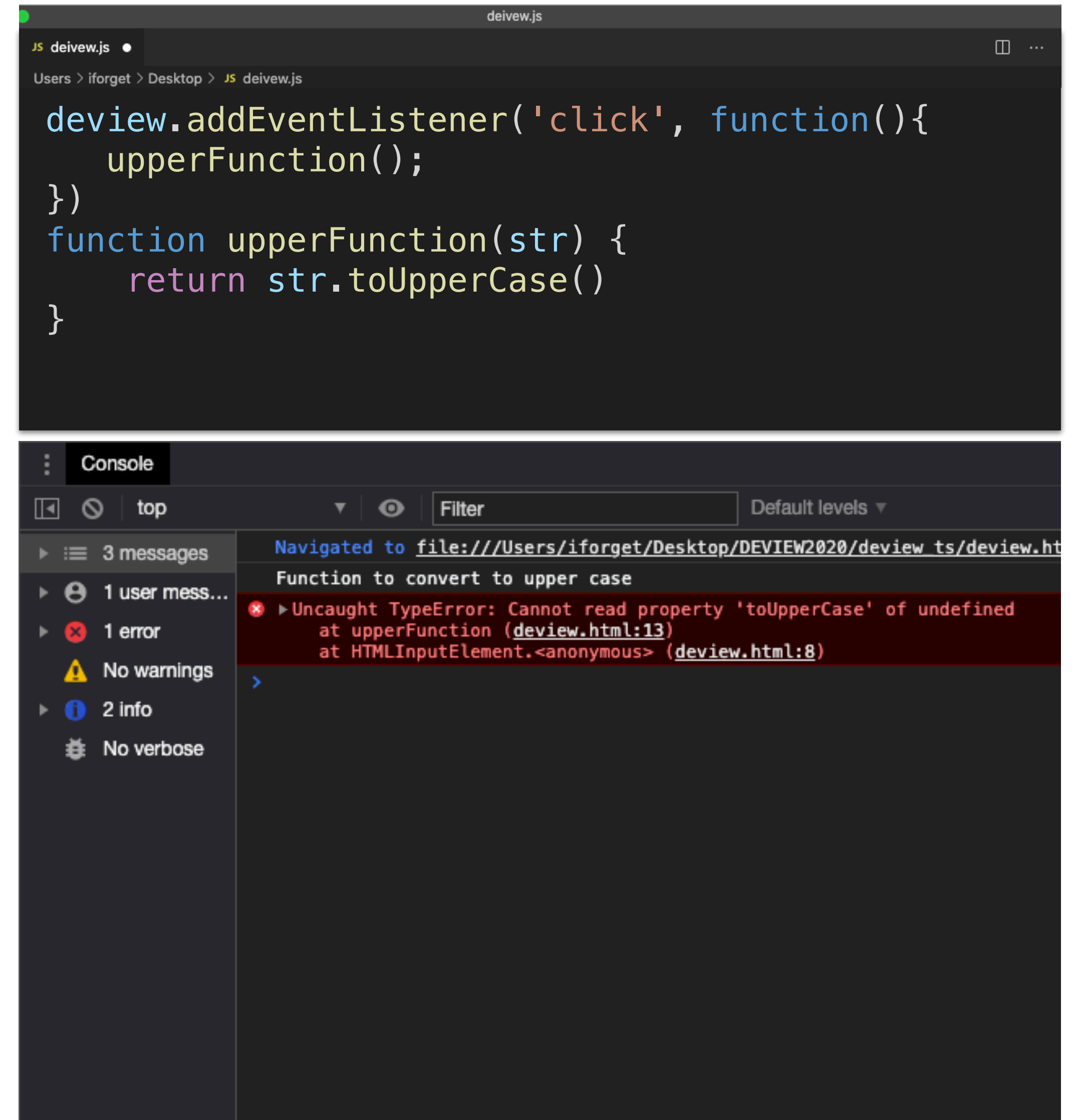
The console sidebar on the left indicates 1 message, 1 error, and no warnings or info.

Need :: Human Error

HOXY 이런 에러들을 본 적 없으신
가요?

- 타입을 잘못 지정하거나
- 전달된 값이 없거나

흔히하는 🤖 Human Error



The screenshot shows a web browser window with a dark theme. The address bar shows the file path: `file:///Users/forget/Desktop/DEVVIEW2020/devview ts/devview.ht`. The page content displays a JavaScript snippet in a monospace font:

```
devview.addEventListener('click', function(){
  upperFunction();
})
function upperFunction(str) {
  return str.toUpperCase()
}
```

Below the code, the browser's console is open, showing a list of messages on the left and the details of an error on the right. The error message is:

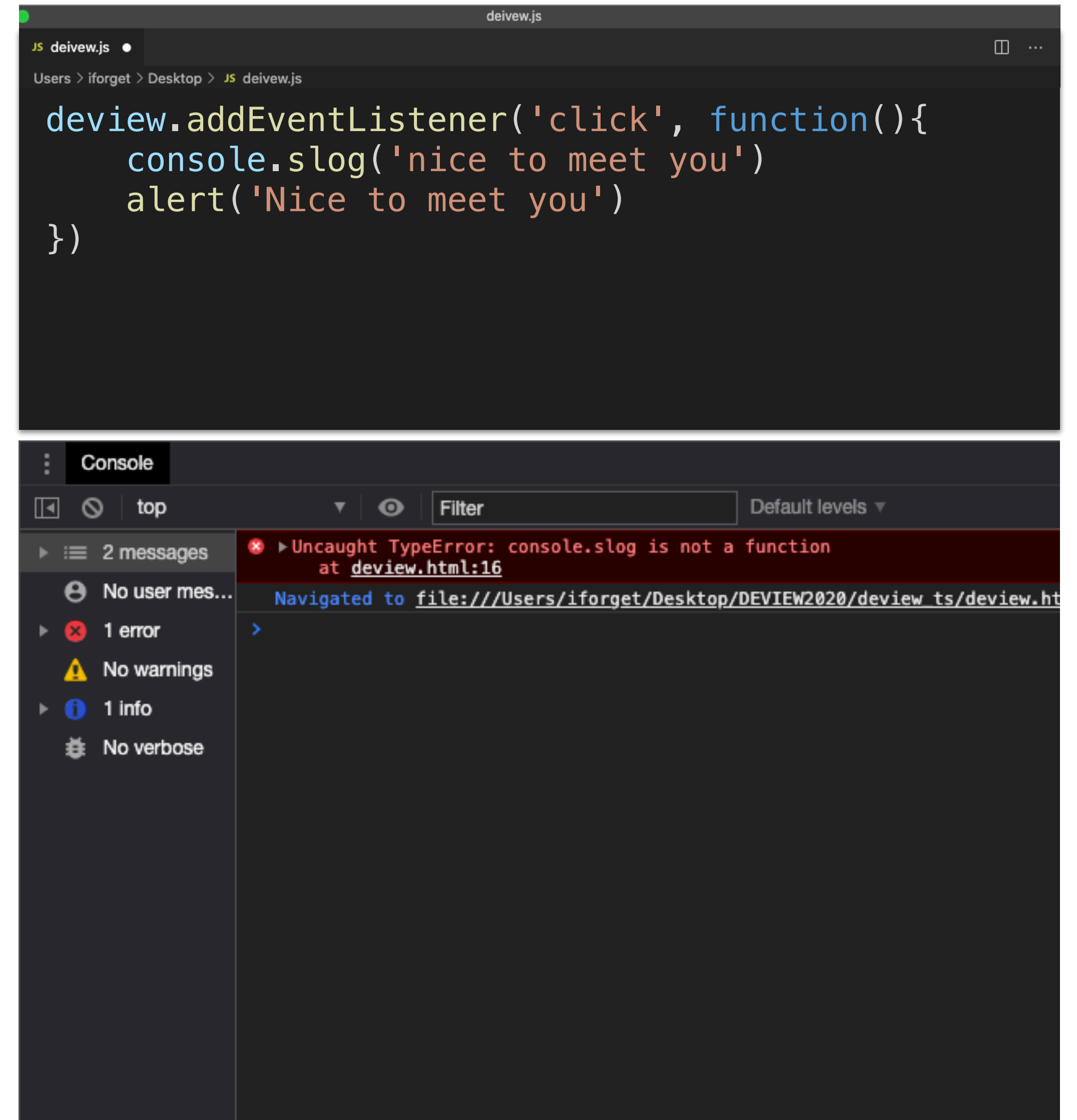
```
Uncaught TypeError: Cannot read property 'toUpperCase' of undefined
    at upperFunction (devview.html:13)
    at HTMLInputElement.<anonymous> (devview.html:8)
```

Need :: Human Error

HOXY 이런 에러들을 본 적 없으신
가요?

- 타입을 잘못 지정하거나
- 전달된 값이 없거나
- 오타자를 치거나

흔히하는 🤖 Human Error



The screenshot shows a web browser window with a dark theme. The address bar shows the file path: `file:///Users/forget/Desktop/DEVVIEW2020/devview ts/devview.ht`. The browser's developer console is open, displaying an error message: `Uncaught TypeError: console.slog is not a function` at `devview.html:16`. The error message is in red text on a dark background. The console also shows a navigation message: `Navigated to file:///Users/forget/Desktop/DEVVIEW2020/devview ts/devview.ht`. The browser's address bar shows the file path: `file:///Users/forget/Desktop/DEVVIEW2020/devview ts/devview.ht`. The browser's developer console is open, displaying an error message: `Uncaught TypeError: console.slog is not a function` at `devview.html:16`. The error message is in red text on a dark background. The console also shows a navigation message: `Navigated to file:///Users/forget/Desktop/DEVVIEW2020/devview ts/devview.ht`.

```
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

devview.addEventListener('click', function(){
  console.slog('nice to meet you')
  alert('Nice to meet you')
})
```

Console

2 messages

No user mes...

1 error

No warnings

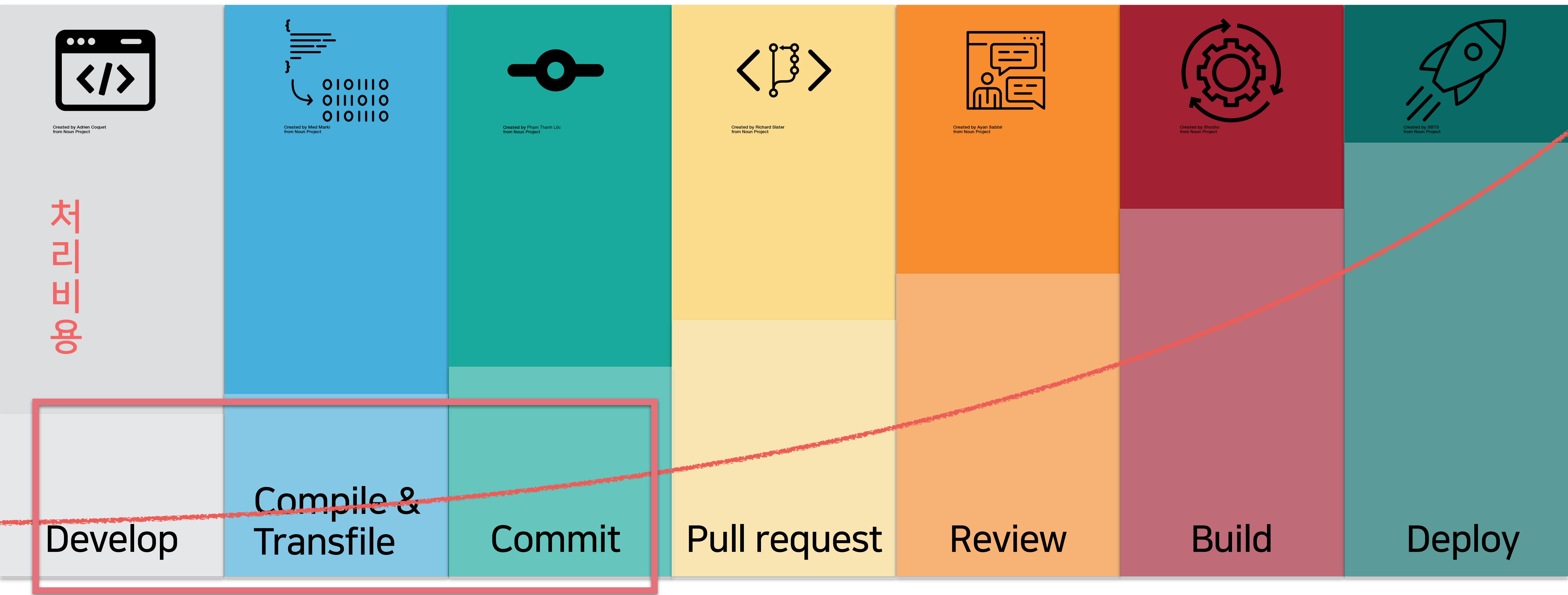
1 info

No verbose

Uncaught TypeError: console.slog is not a function
at devview.html:16

Navigated to file:///Users/forget/Desktop/DEVVIEW2020/devview ts/devview.ht

Need :: Viewpoint

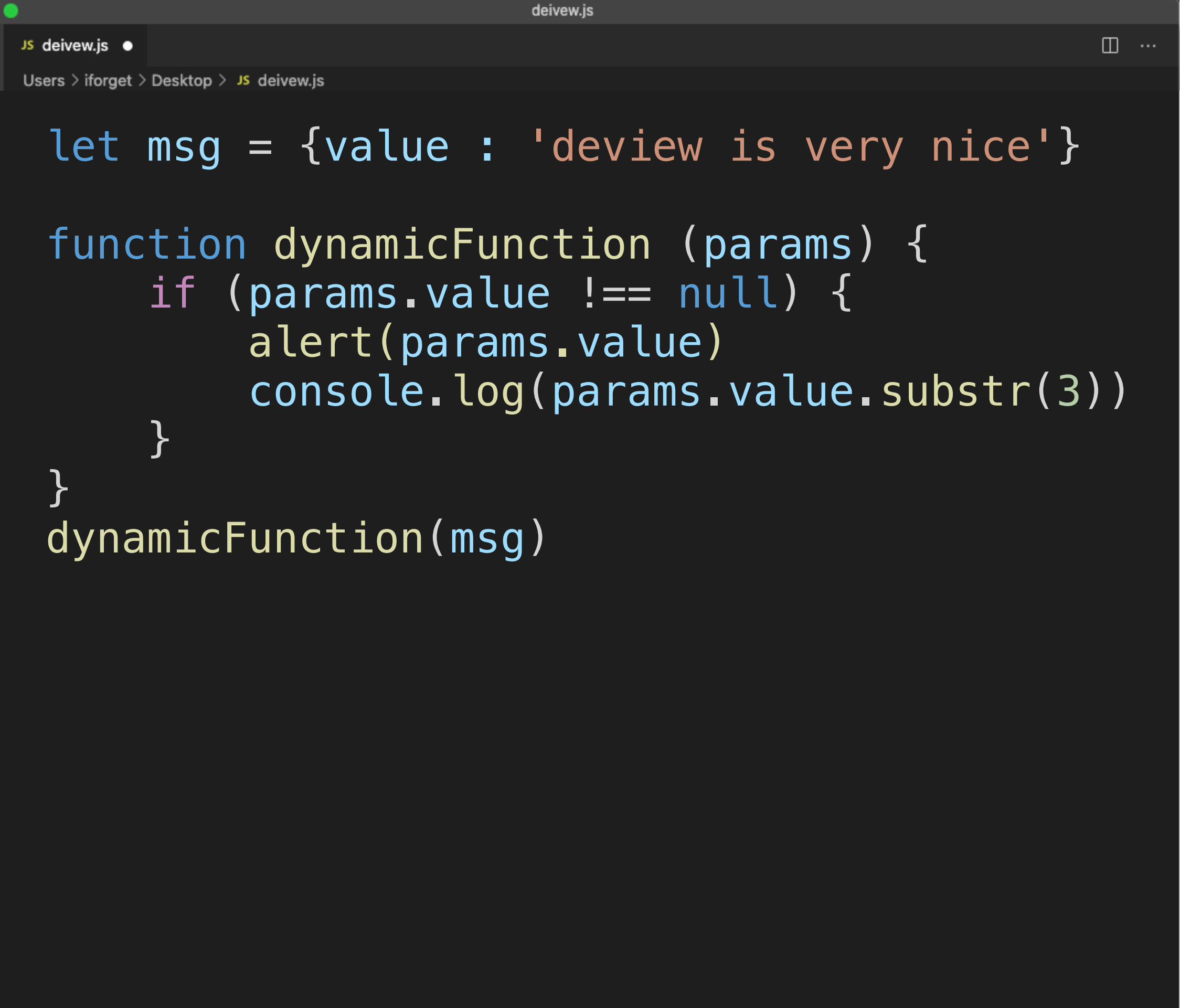


처리비용

버그 발견시점

Advantages of TypeScript

이 코드의 동작은 어떻게 될까요?



```
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

let msg = {value : 'deviiew is very nice'}

function dynamicFunction (params) {
  if (params.value !== null) {
    alert(params.value)
    console.log(params.value.substr(3))
  }
}

dynamicFunction(msg)
```

Advantages of TypeScript

이 코드의 동작은 어떻게 될까요?

- 사실은 아무도 알수 없습니다.

```
deivew.js
JS deivew.js •
Users > iforget > Desktop > JS deivew.js

let msg = {value : 'devview is very nice'}

function dynamicFunction (params) {
  if (params.value !== null) {
    alert(params.value)
    console.log(params.value.substr(3))
  }
}

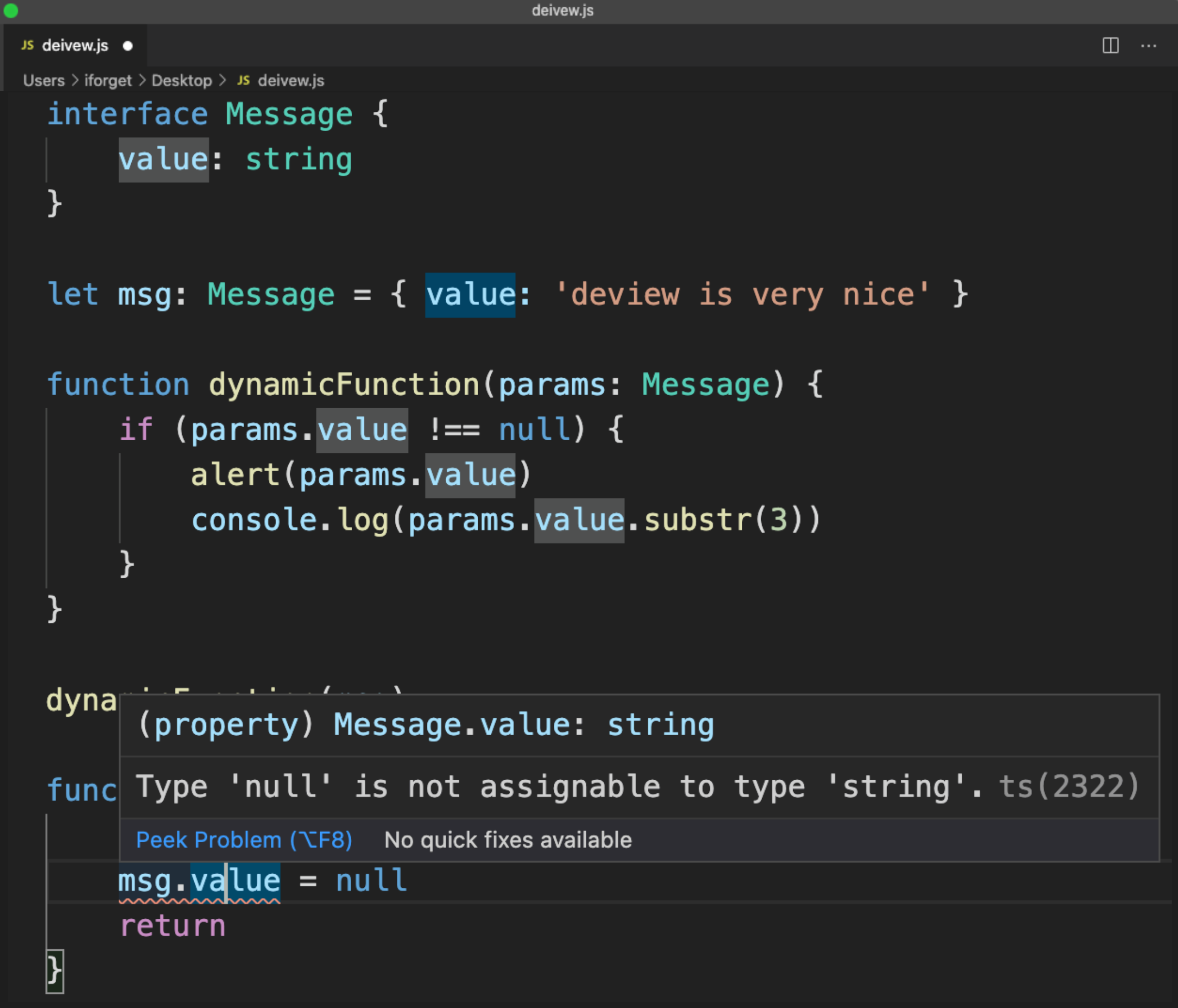
dynamicFunction(msg)


function alert(str) {
  str += '!'
  msg.value = null
  return
}
```


Advantages of TypeScript

이 코드의 동작은 어떻게 될까요?

- 사실은 아무도 알수 없습니다.



```
JS deivew.js •
Users > iforget > Desktop > JS deivew.js
interface Message {
  value: string
}

let msg: Message = { value: 'devview is very nice' }

function dynamicFunction(params: Message) {
  if (params.value !== null) {
    alert(params.value)
    console.log(params.value.substr(3))
  }
}

dynamicFunction(msg)

msg.value = null
return
```

(property) Message.value: string

func Type 'null' is not assignable to type 'string'. ts(2322)

Peek Problem (⌘F8) No quick fixes available

Static Type Checking

```
devview.ts — devview_ts  
TS devview.ts •  
  
const eventBtn = document.getElementById('event')  
if (eventBtn) {  
  eventBtn.addEventListener('click', (e) => {  
    const value = power(10)  
    document.getElementById('result').innerText(value)  
  })  
}  
  
function power(value) {  
  const baseValue = document.getElementById('.base').value  
  
  return Math.power(baseValue, value)  
}
```

Static Type Checking

```
devview.ts — devview_ts
TS devview.ts •

const eventBtn = document.getElementById('event')
if (eventBtn) {
  eventBtn.addEventListener('click', (e) => {
    const value = power(10)
    document.getElementById('result').innerText(value)
  })
}

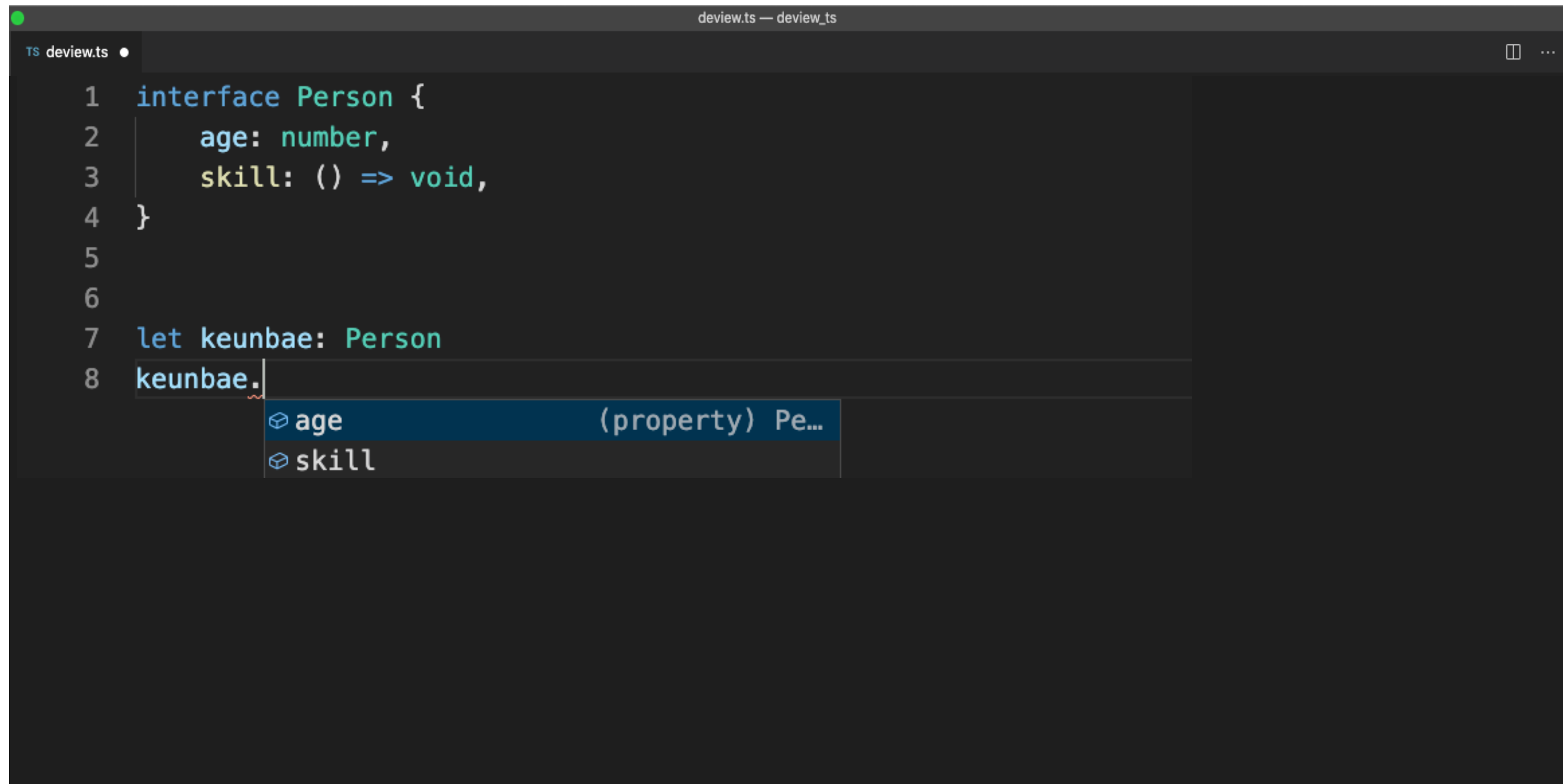
function power(value) {
  const baseValue = document.getElementById('.base').value

  return Math.power(baseValue, value)
}
```


Static Type Checking

```
devview.ts — devview_ts  
TS devview.ts •  
  
const eventBtn = document.getElementById('event')  
if (eventBtn) {  
  eventBtn.addEventListener('click', (e:Event) => {  
    const value = power(10)  
    document.getElementById('result').innerText = String(value)  
  })  
}  
  
function power(value:number) {  
  const baseValue = Number(  
    document.getElementById('.base') as HTMLInputElement).value)  
  
  return Math.pow(baseValue, value)  
}
```

Tooling



The screenshot shows a code editor window titled "devview.ts — devview_ts". The editor contains the following TypeScript code:

```
1 interface Person {  
2     age: number,  
3     skill: () => void,  
4 }  
5  
6  
7 let keunbae: Person  
8 keunbae.
```

An autocomplete dropdown menu is visible below the cursor on line 8, showing two options:

- age (property) Pe...
- skill

3. Node.js

Javascript out of the browser

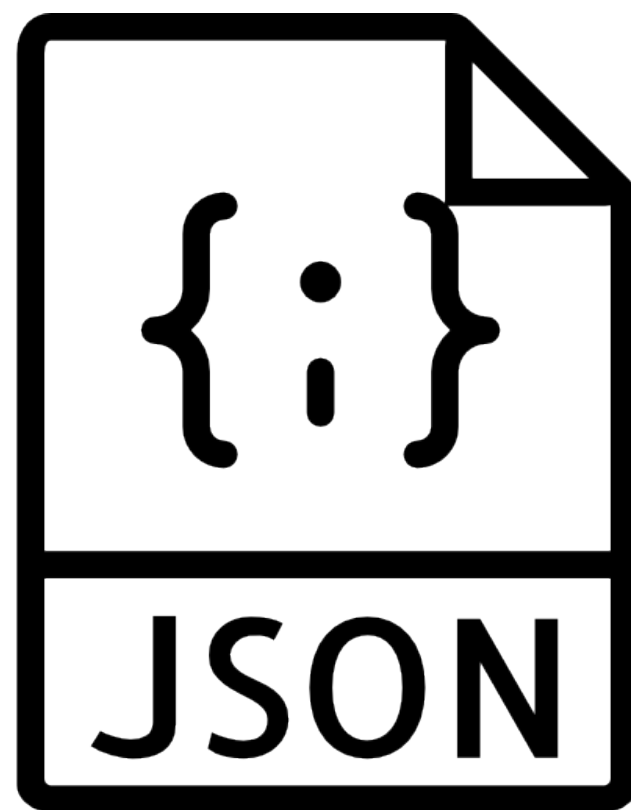


Node.js Feature

Feature

- 비동기 I/O 처리 / 이벤트 위주
- 빠른속도
- 단일 쓰레드 / 뛰어난 확장성
- 노 버퍼링

How to use Node.js



What is your regret?



[참조] 대한민국 어게인 나훈아 방송 중 캡처

Regret

1. not sticking with Promises
2. security
3. package.json
4. node_modules
5. require("module") without the extension ".js"
6. index.js
7. build system



Not sticking with Promises

`fs.open(path[, flags[, mode]], callback)`

#

► History

- `path` `<string>` | `<Buffer>` | `<URL>`
- `flags` `<string>` | `<number>` See [support of file system flags](#). Default: `'r'`.
- `mode` `<string>` | `<integer>` Default: `0o666` (readable and writable)
- `callback` `<Function>`
 - `err` `<Error>`
 - `fd` `<integer>`

Asynchronous file open. See [open\(2\)](#).

`mode` sets the file mode (permission and sticky bits), but only if the file was created. On Windows, only the write permission can be manipulated; see [fs.chmod\(\)](#).

The callback gets two arguments (`err`, `fd`).

Some characters (`<` `>` `:` `"` `/` `\` `|` `?` `*`) are reserved under Windows as documented by [Naming Files, Paths, and Namespaces](#). Under NTFS, if the filename contains a colon, Node.js will open a file system stream, as described by [this MSDN page](#).

Functions based on `fs.open()` exhibit this behavior as well: `fs.writeFile()`, `fs.readFile()`, etc.

Security



I don't know what to say. #116

 Closed

FallingSnow opened this issue on 21 Nov 2018 · 666 comments



FallingSnow commented on 21 Nov 2018 · edited ▾

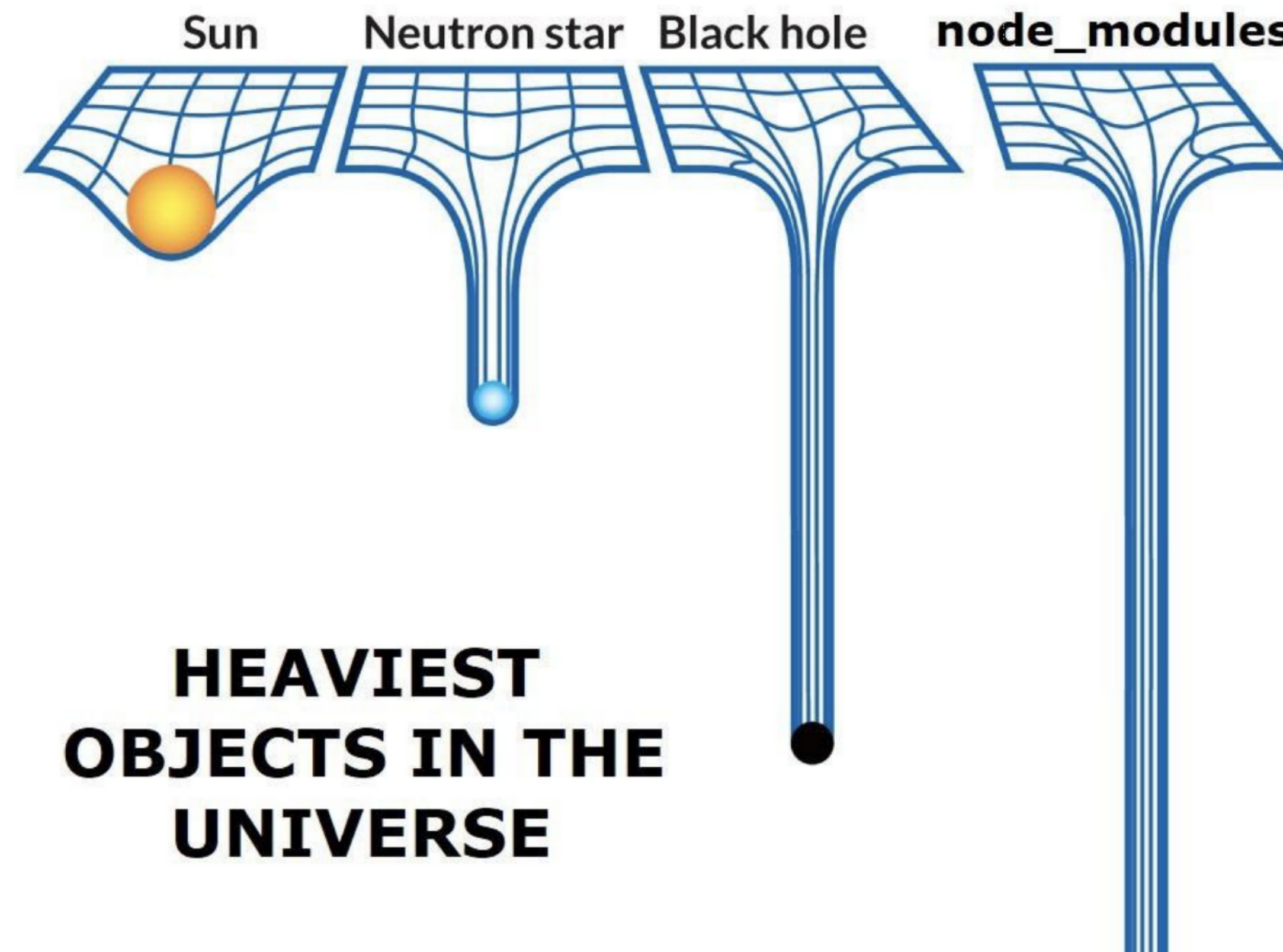
EDIT 26/11/2018:

- Am I affected?:**

If you are using anything crypto-currency related, then maybe. As discovered by @maths22, the target seems to have been identified as copay related libraries. It only executes successfully when a matching package is in use (assumed to be copay at this point). If you are using a crypto-currency related library and if you see `flatmap-stream@0.1.1` after running `npm ls event-stream flatmap-stream`, you are most likely affected. For example:

```
$ npm ls event-stream flatmap-stream
...
flatmap-stream@0.1.1
...
```


node_modules 💔



node_modules 🛑

```
iforget@AD01450034: ~/Works/source/react
└─ master
   └─ zenObservable.js
      └─ zenObservable.js.map
         └─ package.json

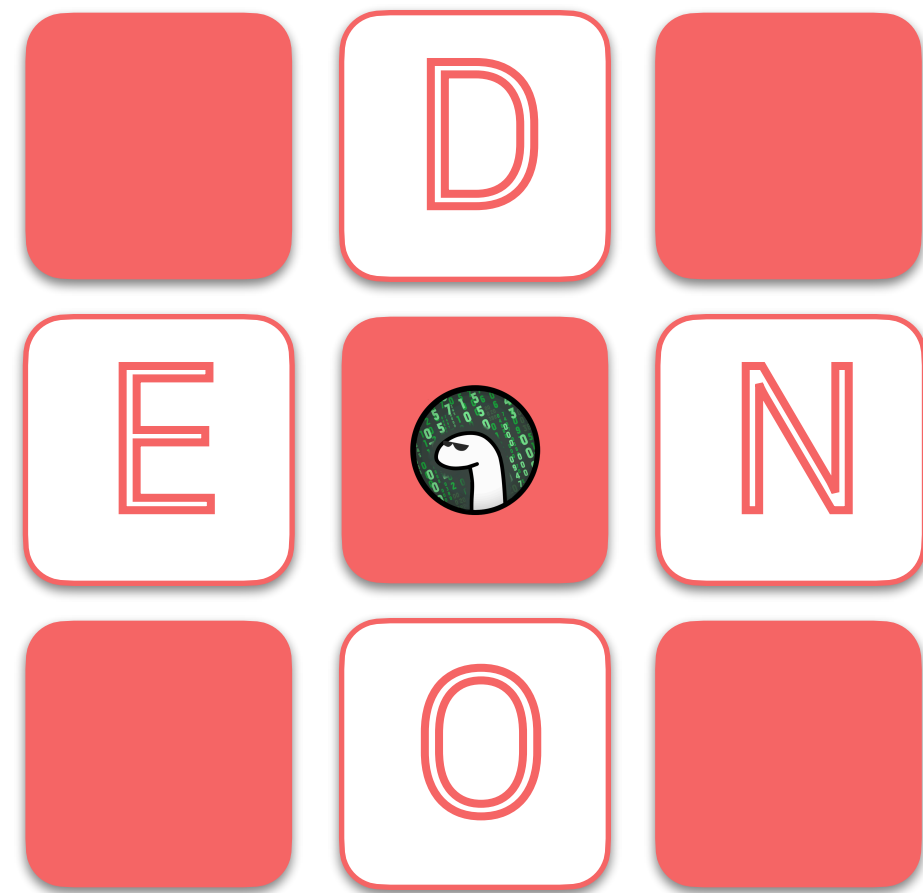
23722 directories, 158495 files
iforget ~/Works/source/react └─ master du -sh ./node_modules
1.5G    ./node_modules
iforget ~/Works/source/react └─ master
```

23,722 directories, 158,495 files

1.5G. ./node_modules

I defined 100 modules

4.About Deno



Feature

A Critical view

Limitations



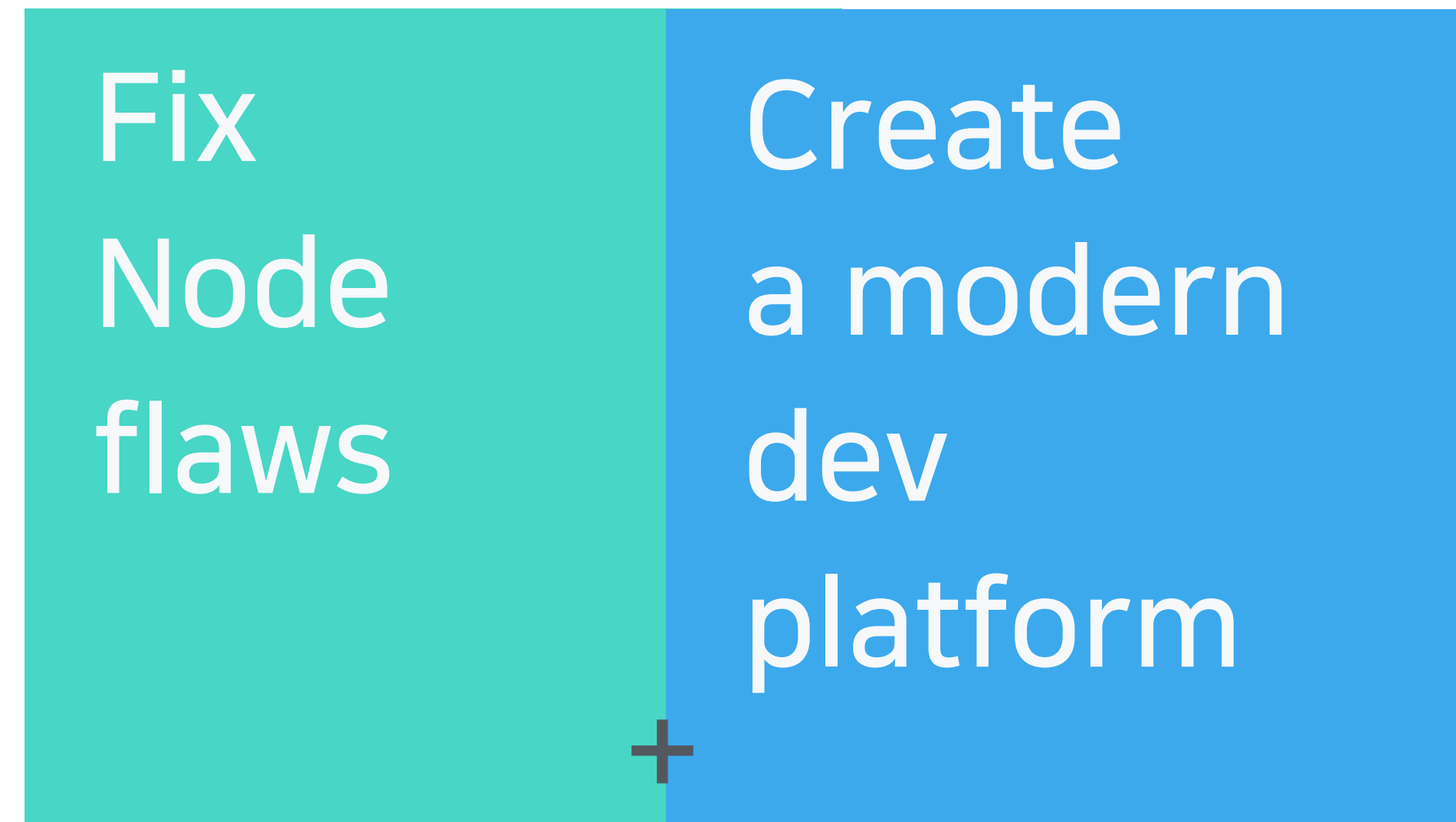
A secure runtime for javascript and typescript

Js/Ts

V8

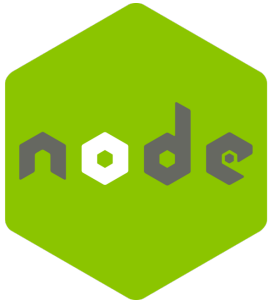
Rust

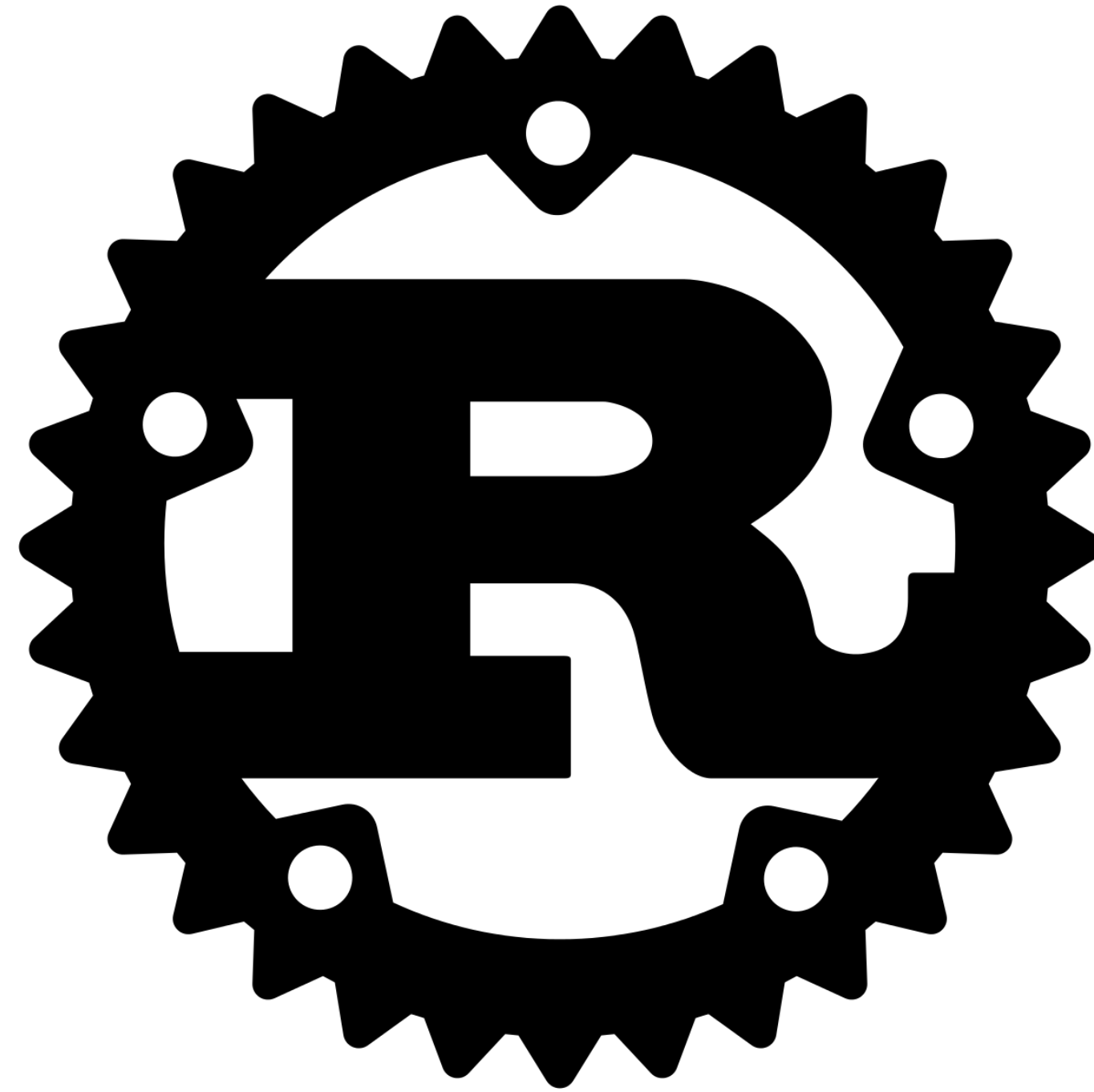
Deno Goals



Deno

Node.js VS Deno

	ENGINE	WRITTEN IN	PACAKAGE MANAGE	IMPORTING PACKAGES	SECURITY	ASYNC PROGRAMING
	V8	C++, JavaScript	npm	CommonJS	Full	Callback
	V8	RUST & TypeScript	uses url	ES Module	Permissioned	Promise

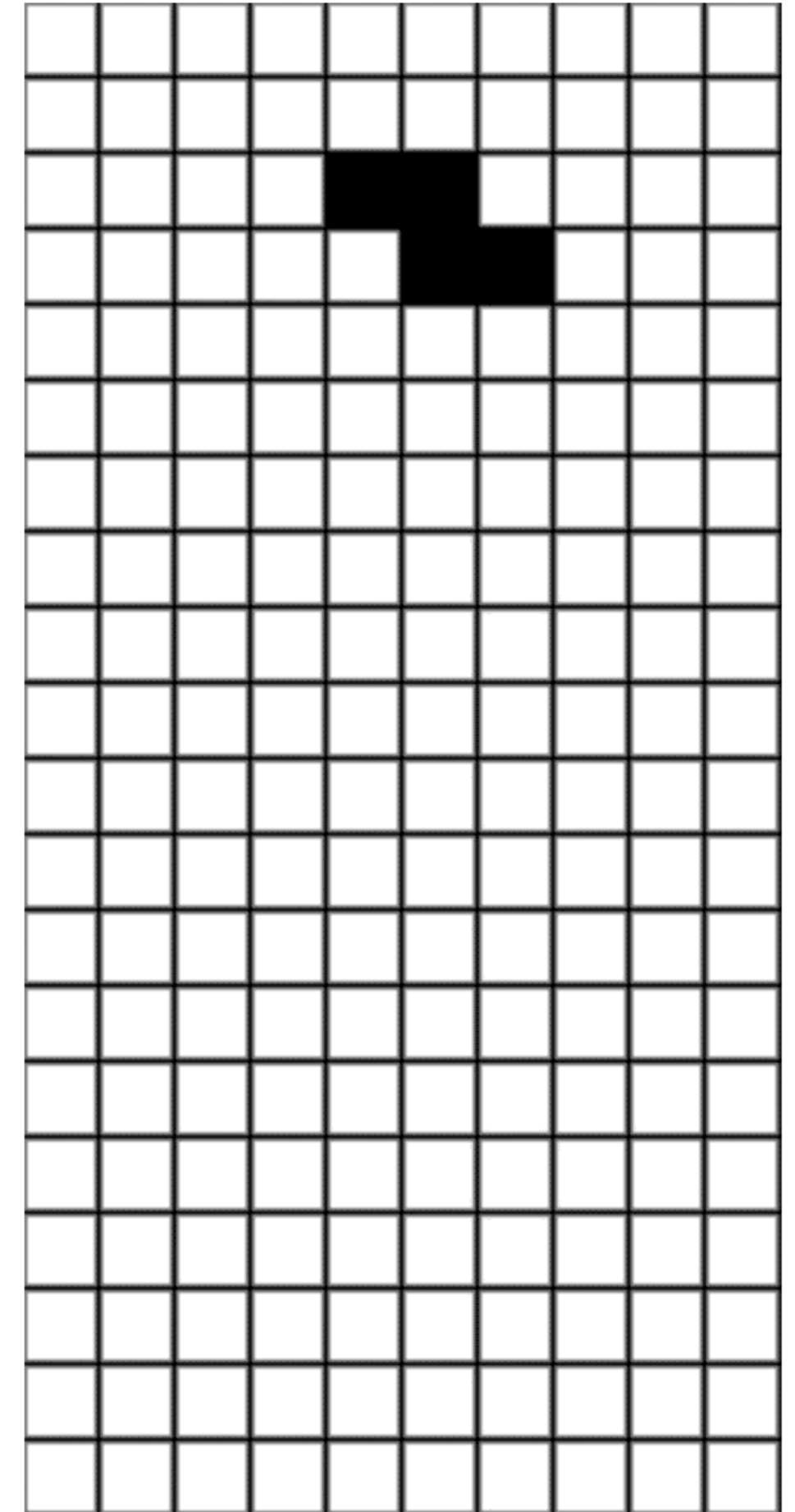


Safe, parallel, practical

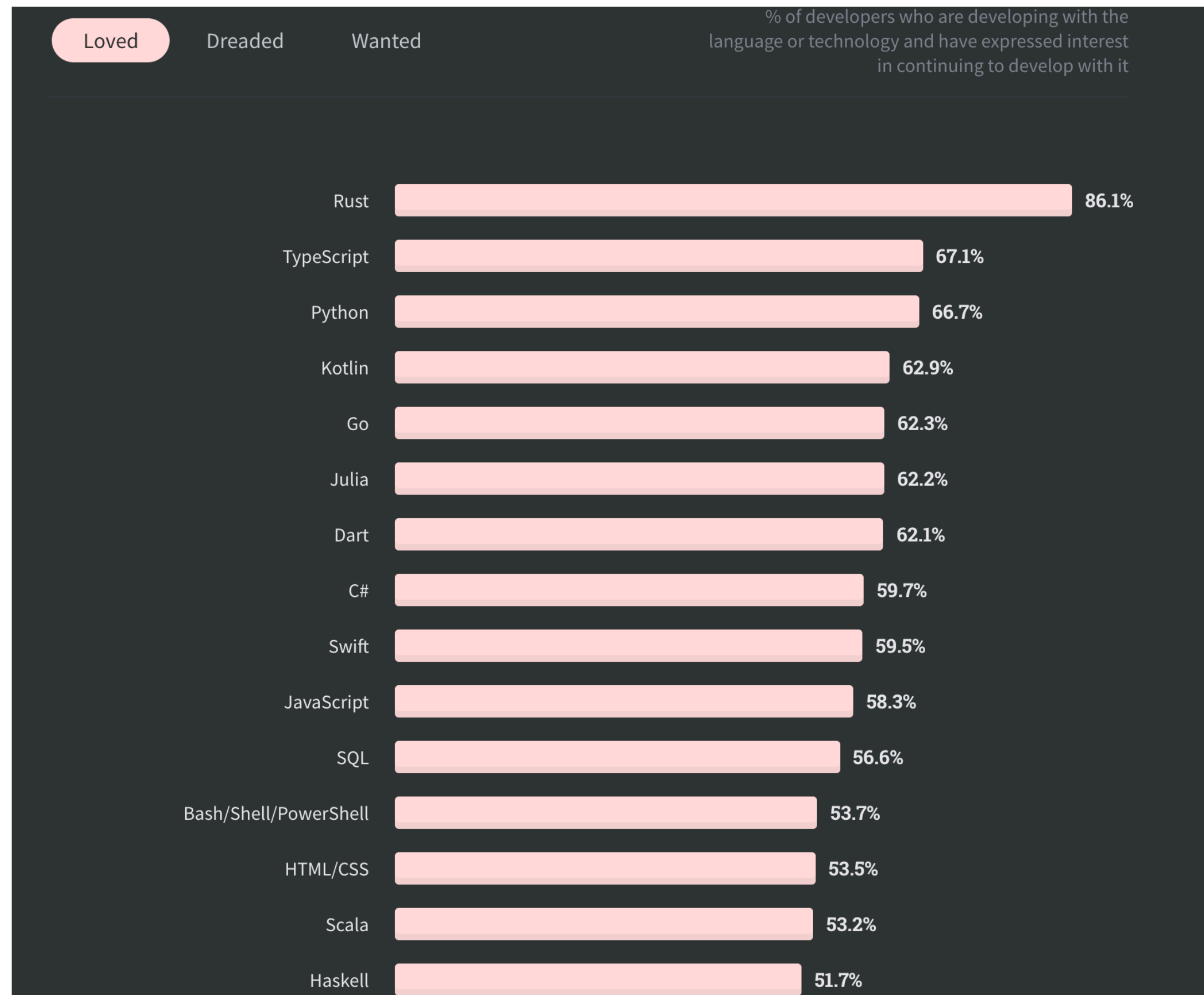
Feature

- 고수준 프로그래밍 언어와 저수준 시스템 제어결합
- C/C++ 처럼 간결한 신택스
- 메모리 / 스레드 / 버그 측면에서 매우 뛰어난 안전성
- 기본 시스템 코드로 컴파일 되어 빠른 속도

Click to Start!



Rust



```
JS deivew.js
Users > iforget > Desktop > JS deivew.js

// max javascript number value = 2147483647.
so maxRounds < sqrt(2147483647) = 46340.95
const scale = 1000000; // nanoseconds
const start = performance.now() * scale;

let count = 0;
if (type == "deno") {
  for (let i = 0; i < rounds; i++) {
    for (let j = 0; j < rounds; j++) {
      count += 1;
    }
  }
} else {
  count = rwasm.compute(rounds);
}

const duration = performance.now() * scale -
start;
```

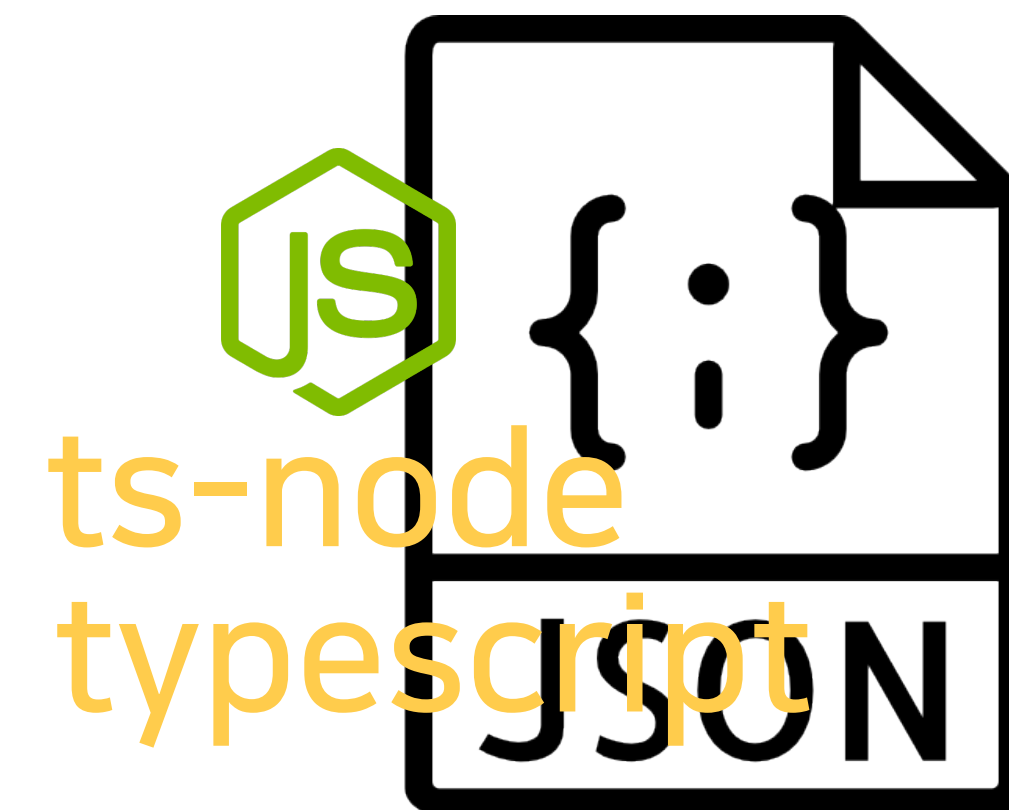

Deno Features

1. Use typescript or javascript
2. Secure by default
3. Decentralized packages
4. Standard Library
5. Modern JS
6. ES Modules
7. Top Level / First Class await
8. Built In Testing
9. Browser Compatible API
10. bundled into a single js file

TypeScript out of the box

즉시 사용 가능한 TypeScript

- TypeScript 컴파일러와 ts-node 같은 다른 툴이 필요하지 않음.
- Deno 실행파일에 TSC 와 V8 이 내장됨.
- TS configuration 은 옵션



Decentralized packages

분산형 패키지

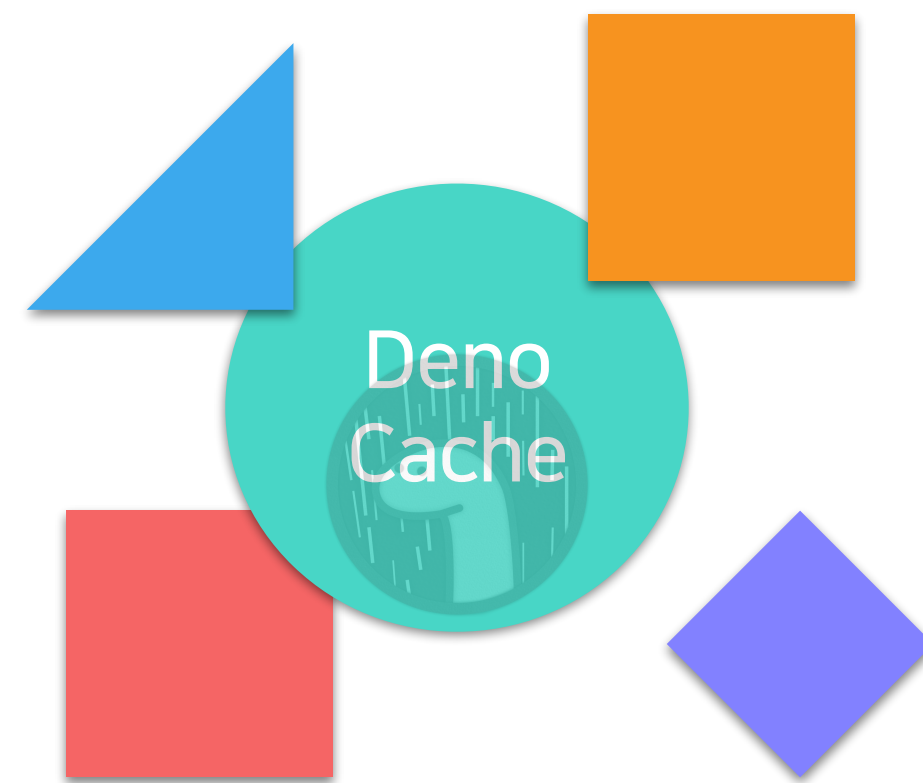
- 더이상 필요 없는 NPM / package.json
- Packages 를 이용하는 방법은 URL - <https://deno.land/x/>

```
import { serve } from 'https://deno.land/std@0.74.0/http/server.ts'
```

Node modules 🤔

노드 모듈

- application 당 node_modules 없음.
- Deno 는 여전히 어딘가에 모듈을 가져오거나 컴파일을 해줘야함.



Node modules 🤔

노드 모듈

- application 당 node_modules 없음.
- Deno 는 여전히 어딘가에 모듈을 가져오거나 컴파일을 해줘야함.

```
$ deno info
```

```
DENO_DIR location: "/Users/deno/Library/Caches/deno"
```

```
Remote modules cache: "/Users/deno/Library/Caches/deno/deps"
```

```
TypeScript compiler cache: "/Users/deno/Library/Caches/deno/gen"
```


보안성

- Node.js 와 달리 Sandbox 에서 코드실행.
- 파일 시스템/인터넷 액세스 하는 앱 실행시 권한 우선 허용.

```
error: Uncaught PermissionDenied: network access to "127.0.0.1:8080", run again with the --allow-net flag
    at processResponse (core.js:226:13)
    at Object.jsonOpSync (core.js:250:12)
    at opListen (deno:cli/rt/30_net.js:32:17)
    at Object.listen (deno:cli/rt/30_net.js:207:17)
    at serve (server.ts:287:25)
    at main (server.ts:6:18)
    at server.ts:14:1
```

Browser Compatible API

브라우저 호환 API

- global 객체 없음. 대신 window 객체 존재

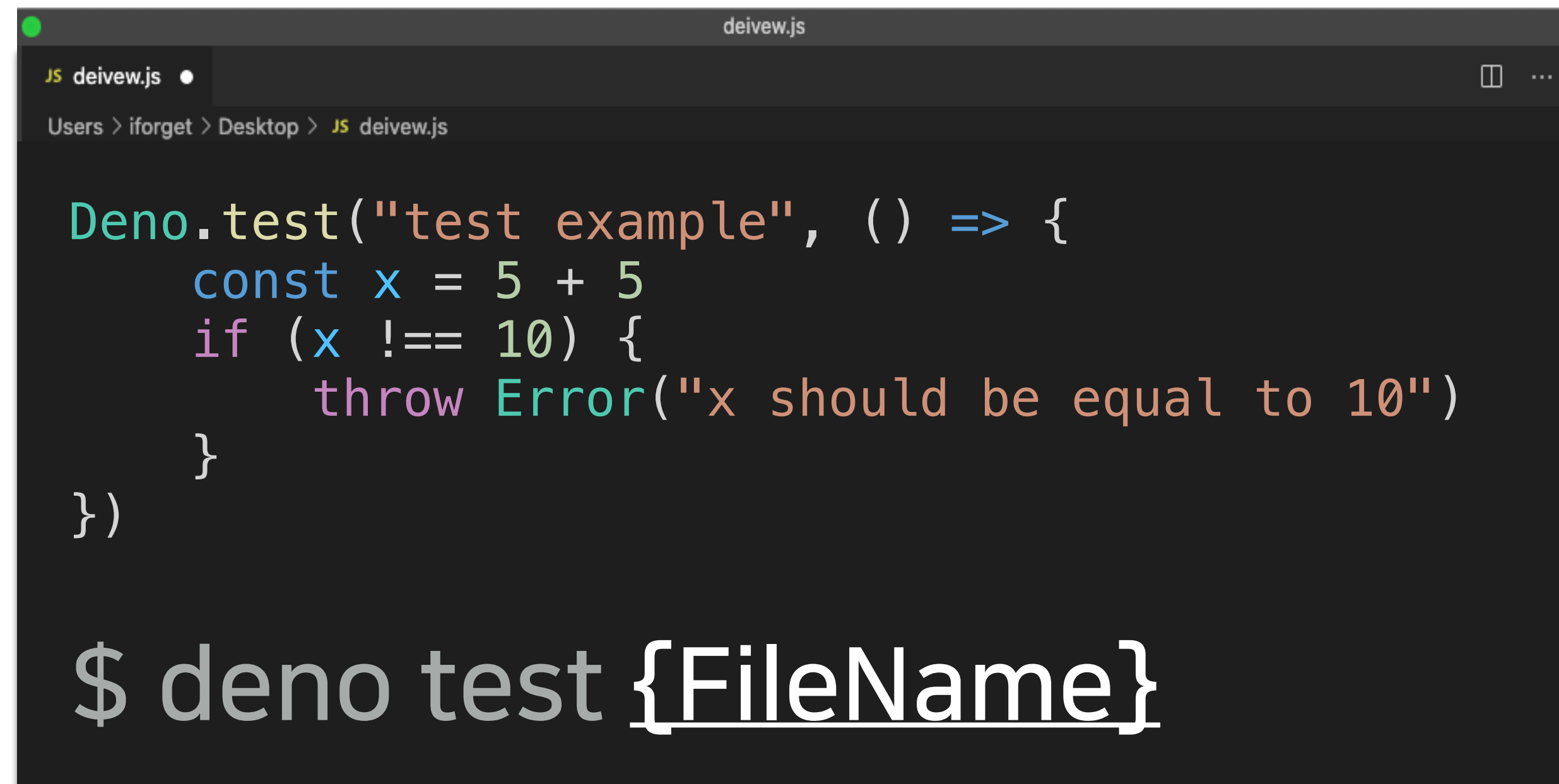


```
iforget ~/Works/source/deno master deno
Deno 1.2.2
exit using ctrl+d or close()
> global
Uncaught ReferenceError: global is not defined
    at <unknown>:3:1
    at evaluate (rt/40_repl.js:60:36)
    at replLoop (rt/40_repl.js:160:15)
> window
{
  console: {
    log: [Function: log],
    debug: [Function: log],
    info: [Function: log],
    dir: [Function: dir],
    dirxml: [Function: dir],
    warn: [Function: warn],
    error: [Function: warn],
    assert: [Function: assert],
    count: [Function: count],
    countReset: [Function: countReset],
    table: [Function: table],
    time: [Function: time],
    timeLog: [Function: timeLog],
    timeEnd: [Function: timeEnd],
```

Built In Testing

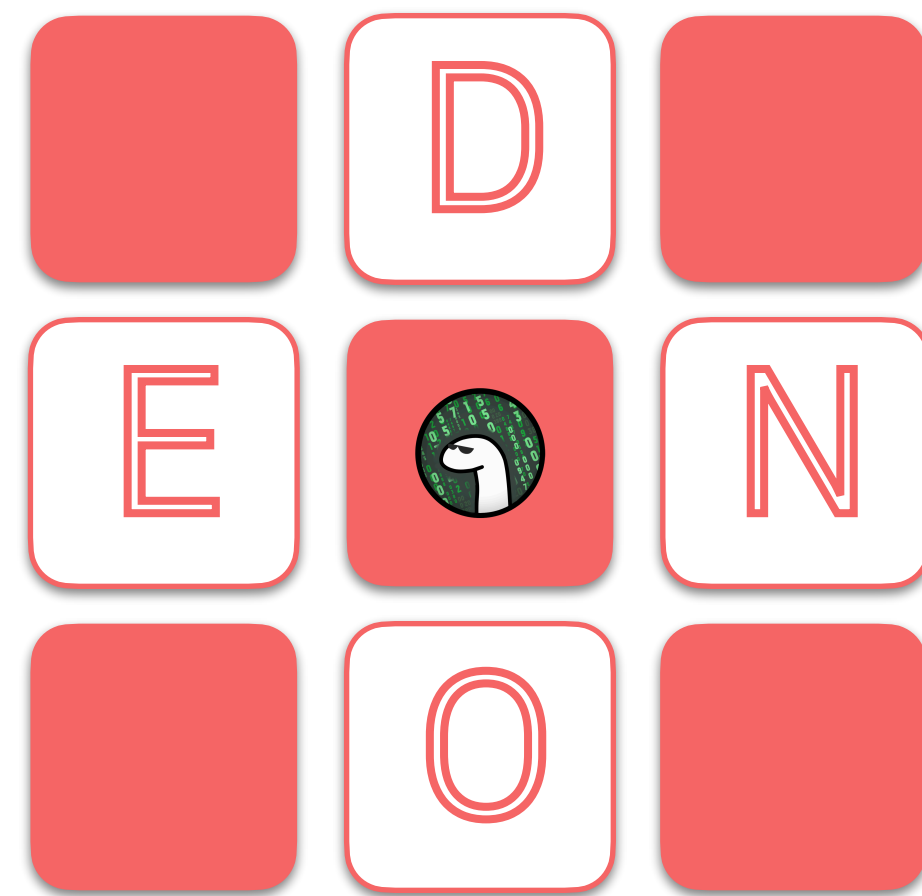
내장 된 테스트

- 자체 내장 테스트 러너 포함.

A screenshot of a code editor window titled 'deivew.js'. The editor shows a JavaScript test function using Deno's test API. The code is as follows:

```
Deno.test("test example", () => {  
  const x = 5 + 5  
  if (x !== 10) {  
    throw Error("x should be equal to 10")  
  }  
})
```

Below the code, the command to run the tests is displayed: `$ deno test {FileName}`. The editor's breadcrumb shows the file path: 'Users > iforget > Desktop > JS deivew.js'.



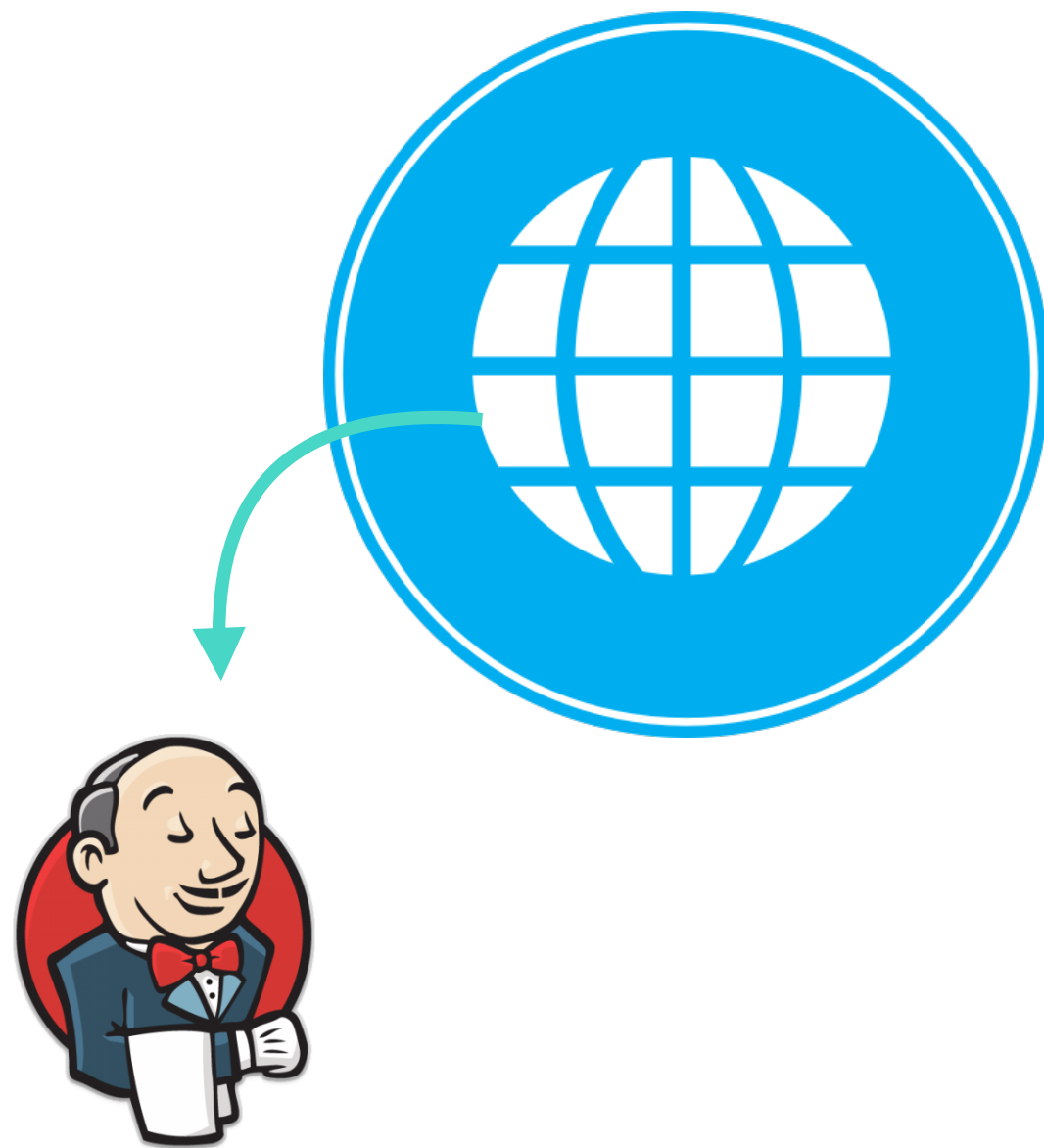
Feature

A critical view

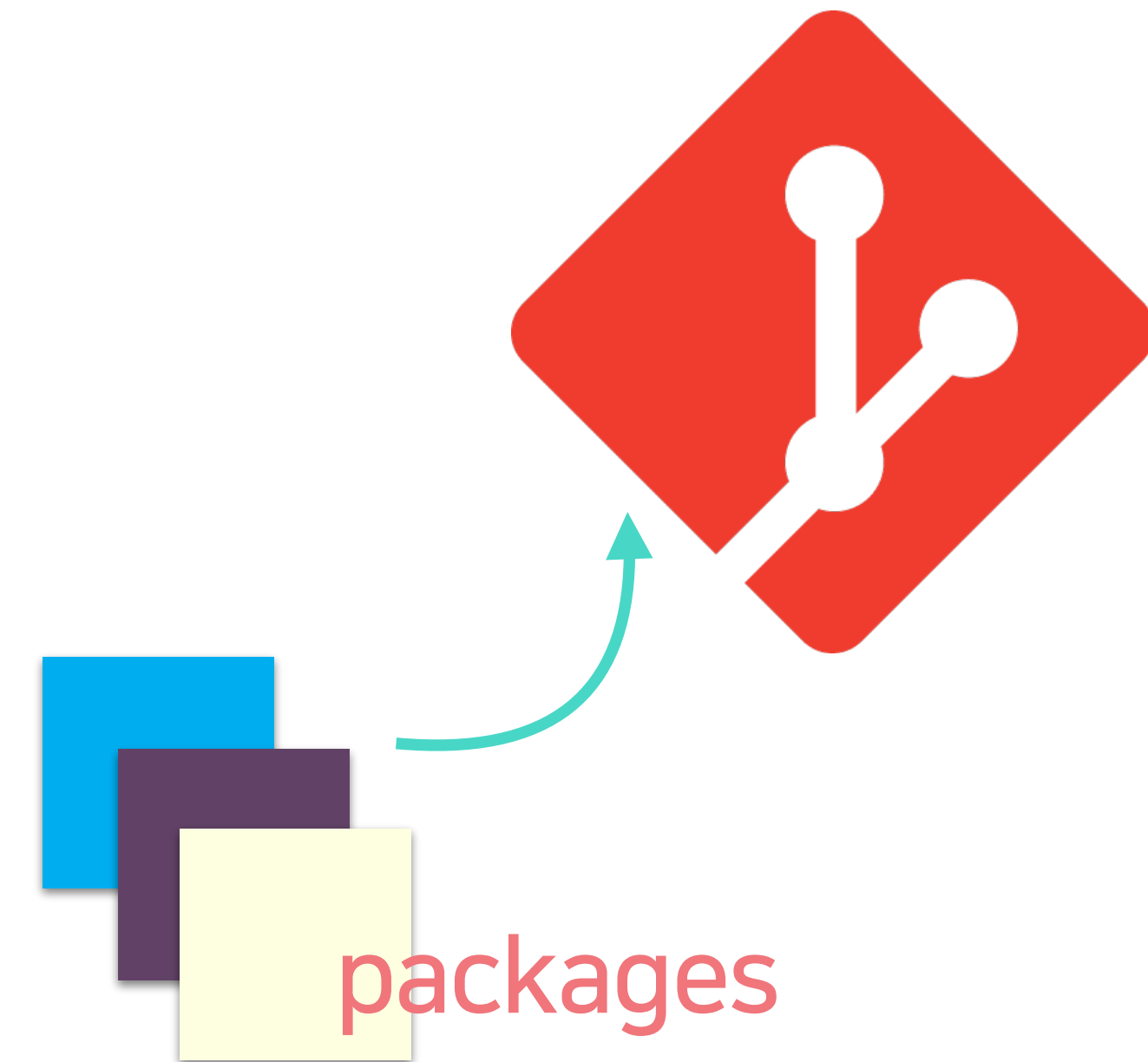
Limitations

Package Manager

Deno 는 CI/CD 와 함께 사용하기 좋지않다.



Case. 1

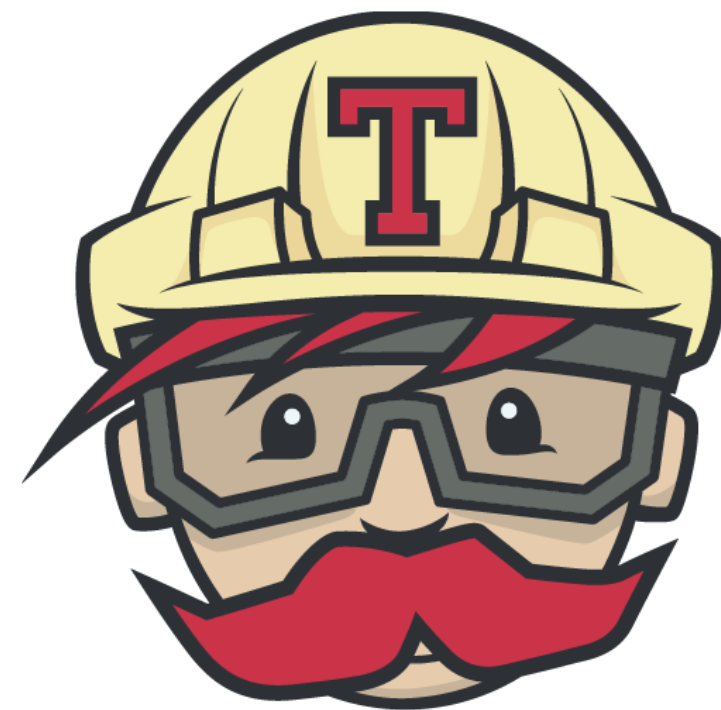


Case. 2

Package Manager

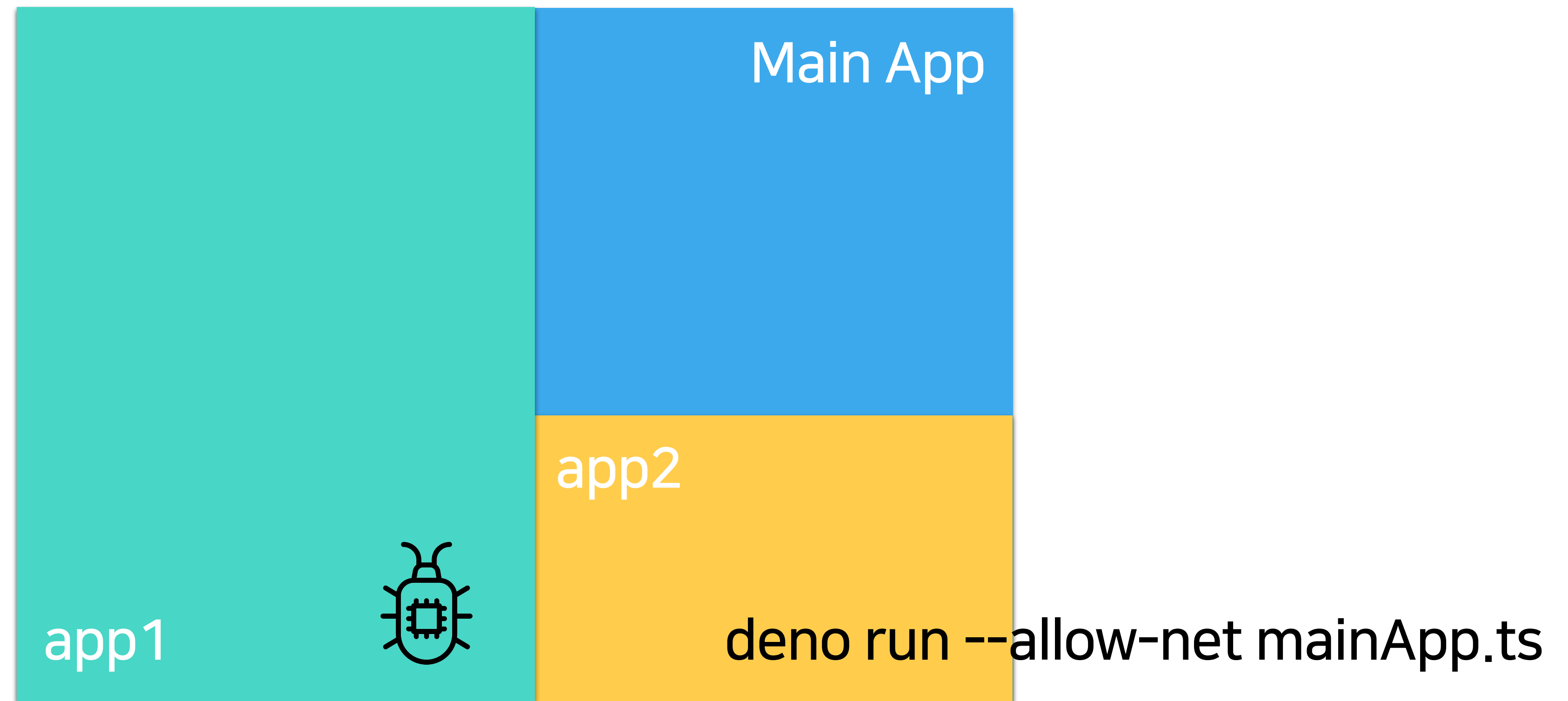
Deno는 CI / CD 배포 성능을 향상시킵니다!

- node_module 접근 방식으로 공통캐시로 대체
- 종속성을 얻기 위해 네트워크 I/O 을 줄이므로 초기화 성능 개선

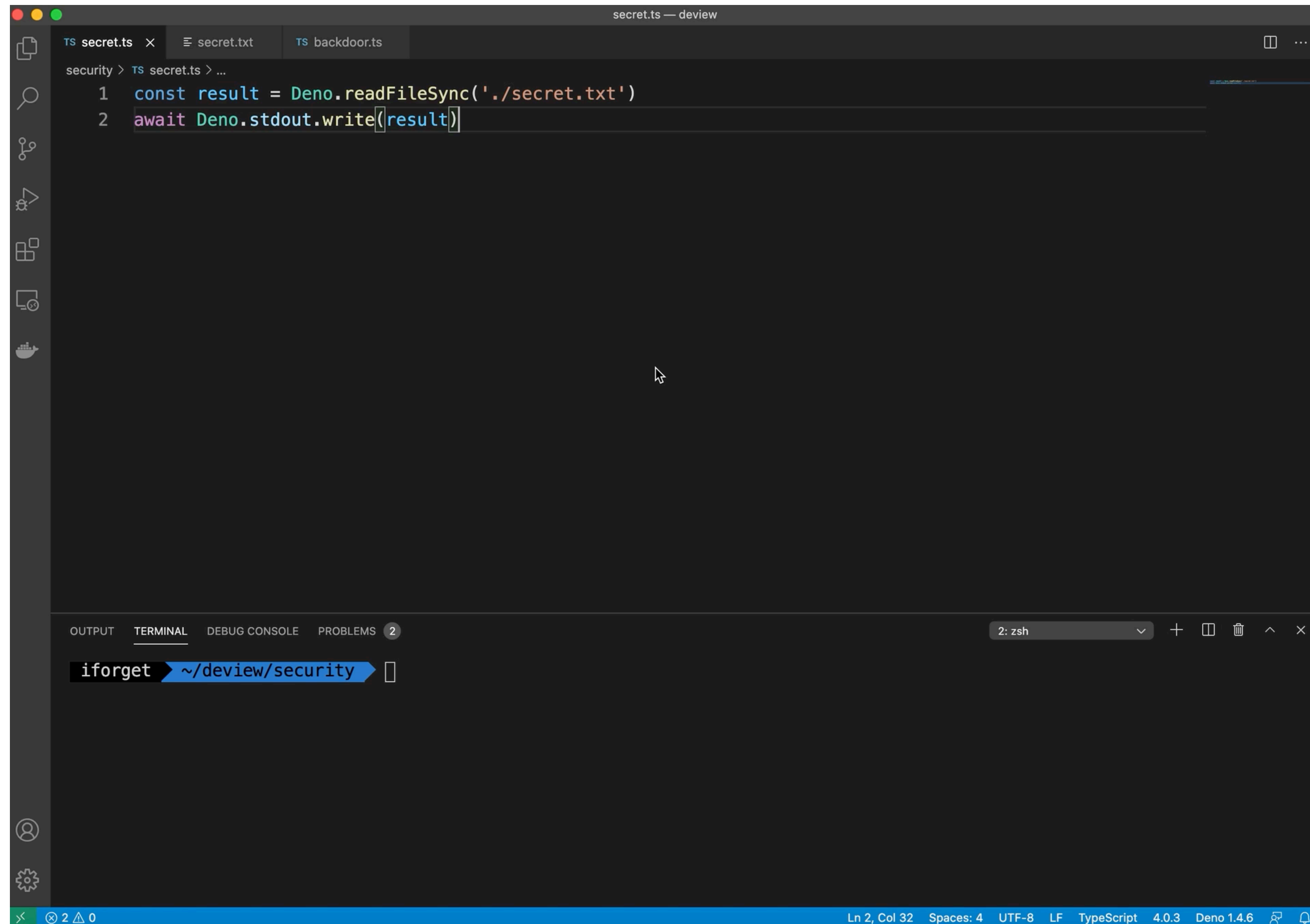


Security policy

보안 정책



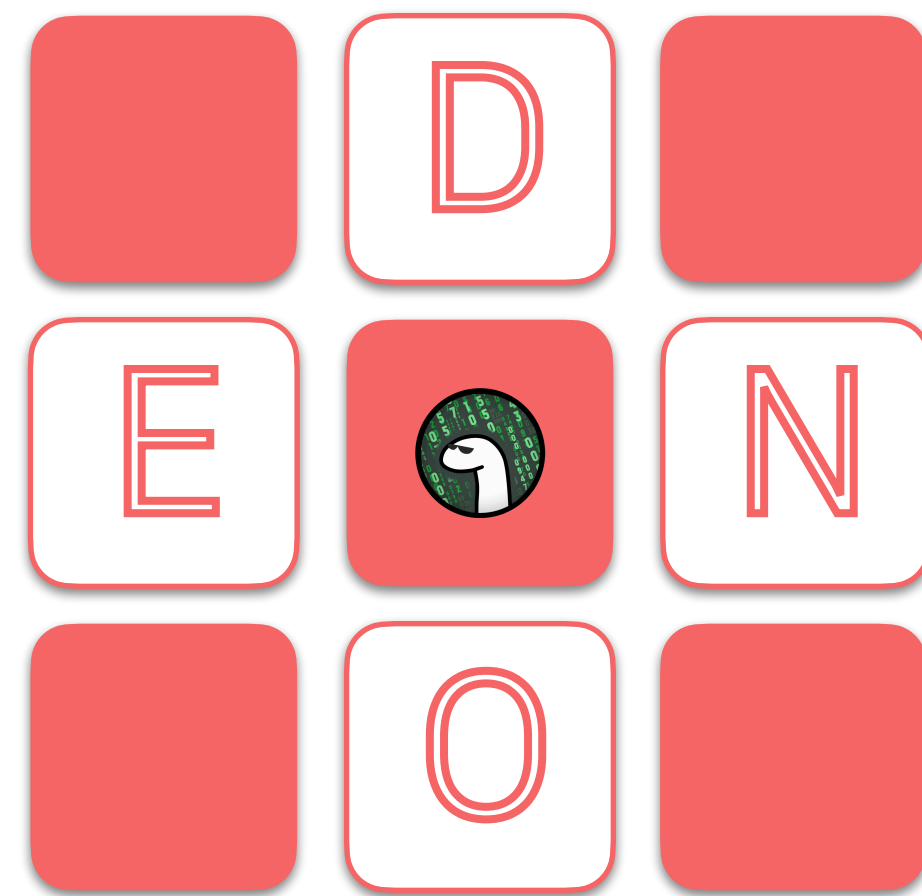
Security policy



The screenshot shows a VS Code editor window titled "secret.ts — devview". The editor has two tabs: "secret.ts" and "TS backdoor.ts". The "secret.ts" tab is active, showing the following code:

```
1 const result = Deno.readFileSync('./secret.txt')
2 await Deno.stdout.write(result)
```

The terminal window at the bottom is titled "2: zsh" and shows the command "iforget ~/devview/security" being executed. The status bar at the bottom indicates the current position is "Ln 2, Col 32", the file is encoded in "UTF-8" with "LF" line endings, and it is a "TypeScript" file using "Deno 1.4.6".



Feature
A Critical view
Limitations

Limitations

호환성

- deno는 대부분의 경우 node.js 패키지와 호환되지 않음.
- 기본 레이어는 Node 와 호환성을 맞추려고 하고 있음.

Deno는 일반적으로 Node 패키지(npm을 이용해 설치 및 관리하는 패키지)와 호환되지 않습니다. 호환성 레이어를 <https://deno.land/std/node/>에 구축하긴 했지만, 완전하지는 않습니다.
출처: <https://kiwsx23.tistory.com/533> [香格里拉]

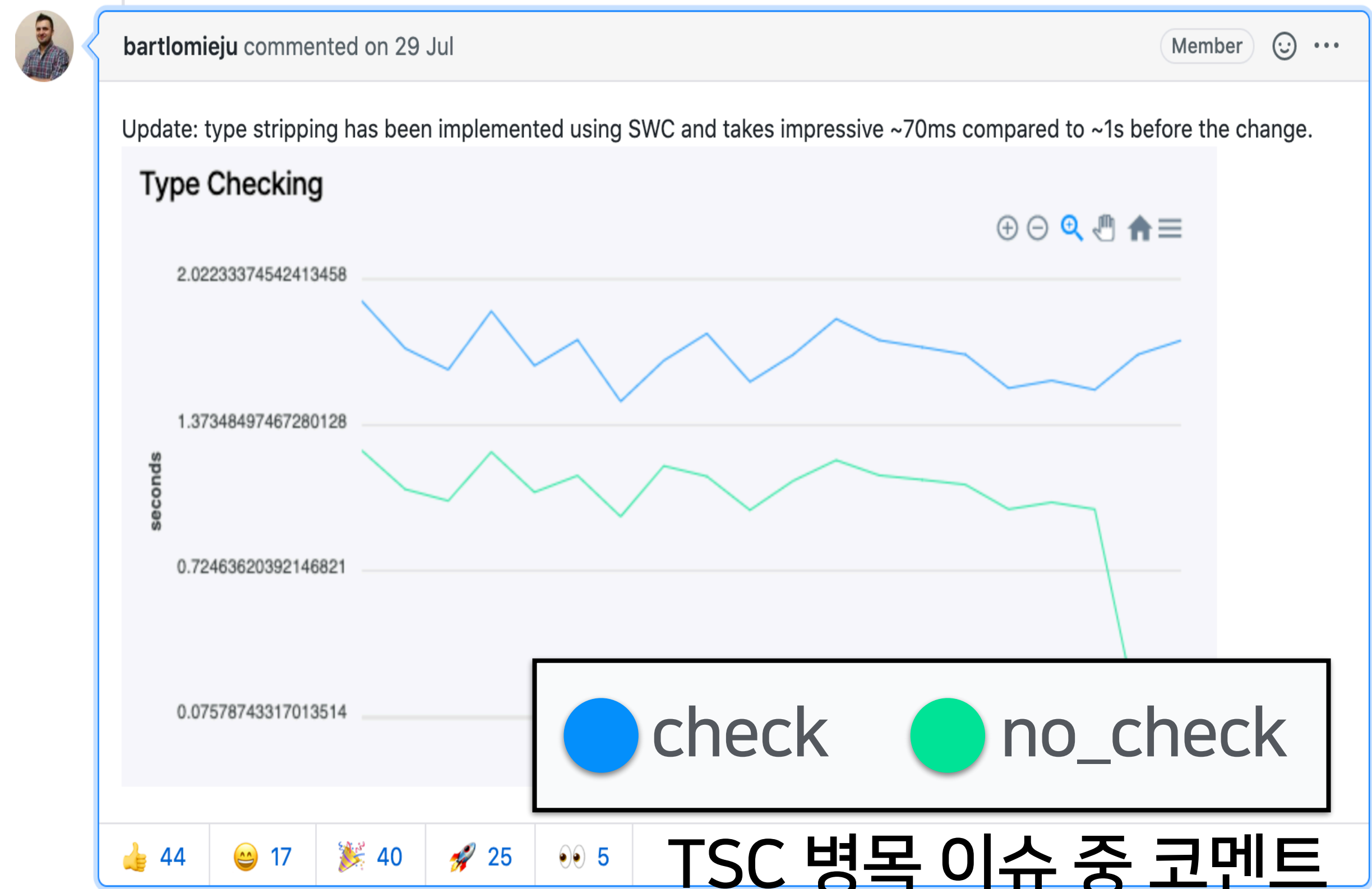
deno.land / std@0.74.0 / node	
 /node	Repository
 Search files...	
 _fs	110 KB
 _util	57 KB
 tests	5 KB
 _errors.ts	16 KB
 _utils.ts	4 KB
 assert_test.ts	2 KB
 assert.ts	649 B
 assertion_error_test.ts	5 KB
 assertion_error.ts	19 KB
 buffer_test.ts	15 KB
 buffer.ts	15 KB
 events_test.ts	18 KB
 events.ts	17 KB
 fs.ts	2 KB
 global.ts	337 B

standard library

Limitations

TSC 병목

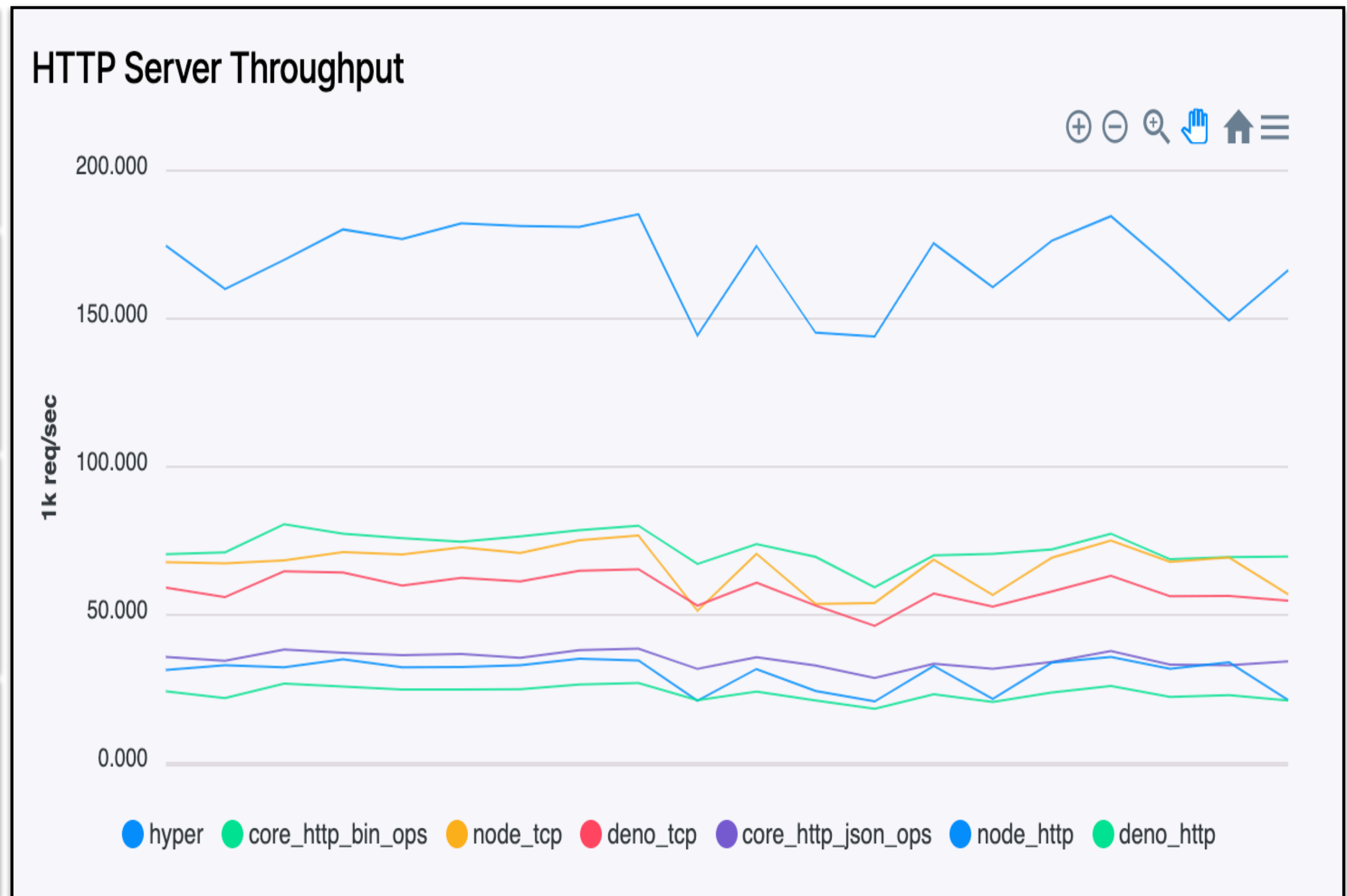
- MS TypeScript Compiler 이용.
- V8 이 JavaScript 코드 파싱보다 상당히 느림.
- 개발자가 체감하는 부분이기에 Rust 로 포팅 예정.



Limitations

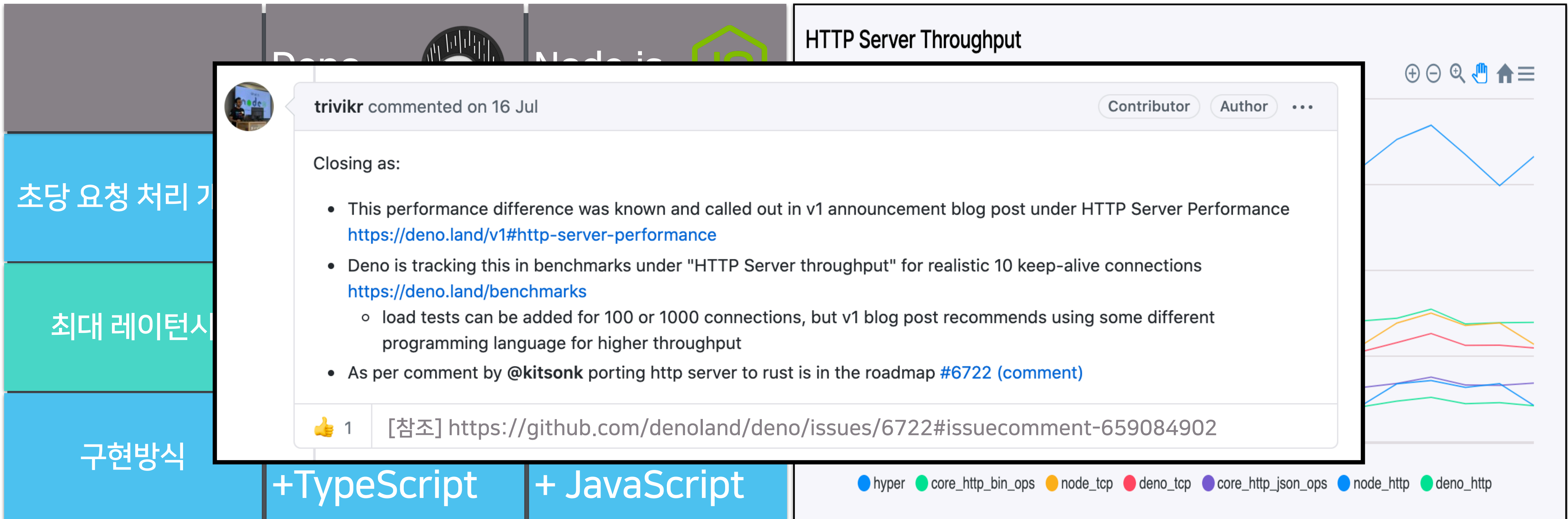
HTTP 서버성능

	Deno 	Node.js 
초당 요청 처리 개수	25,000 req	34,000 req
최대 레이턴시	1.3ms	200~300ms
구현방식	Native tcp +TypeScript	C + JavaScript



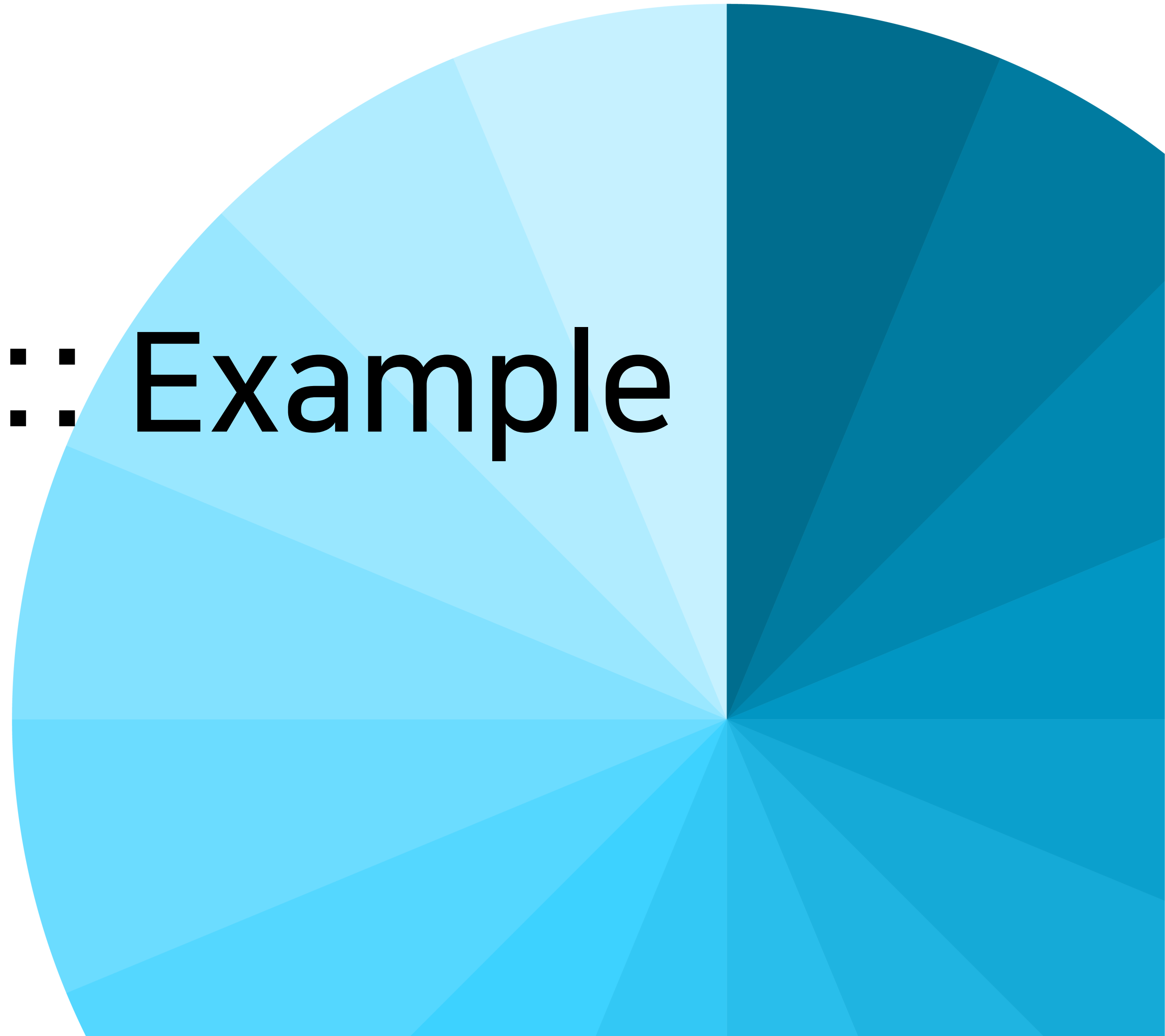
Limitations

HTTP 서버성능

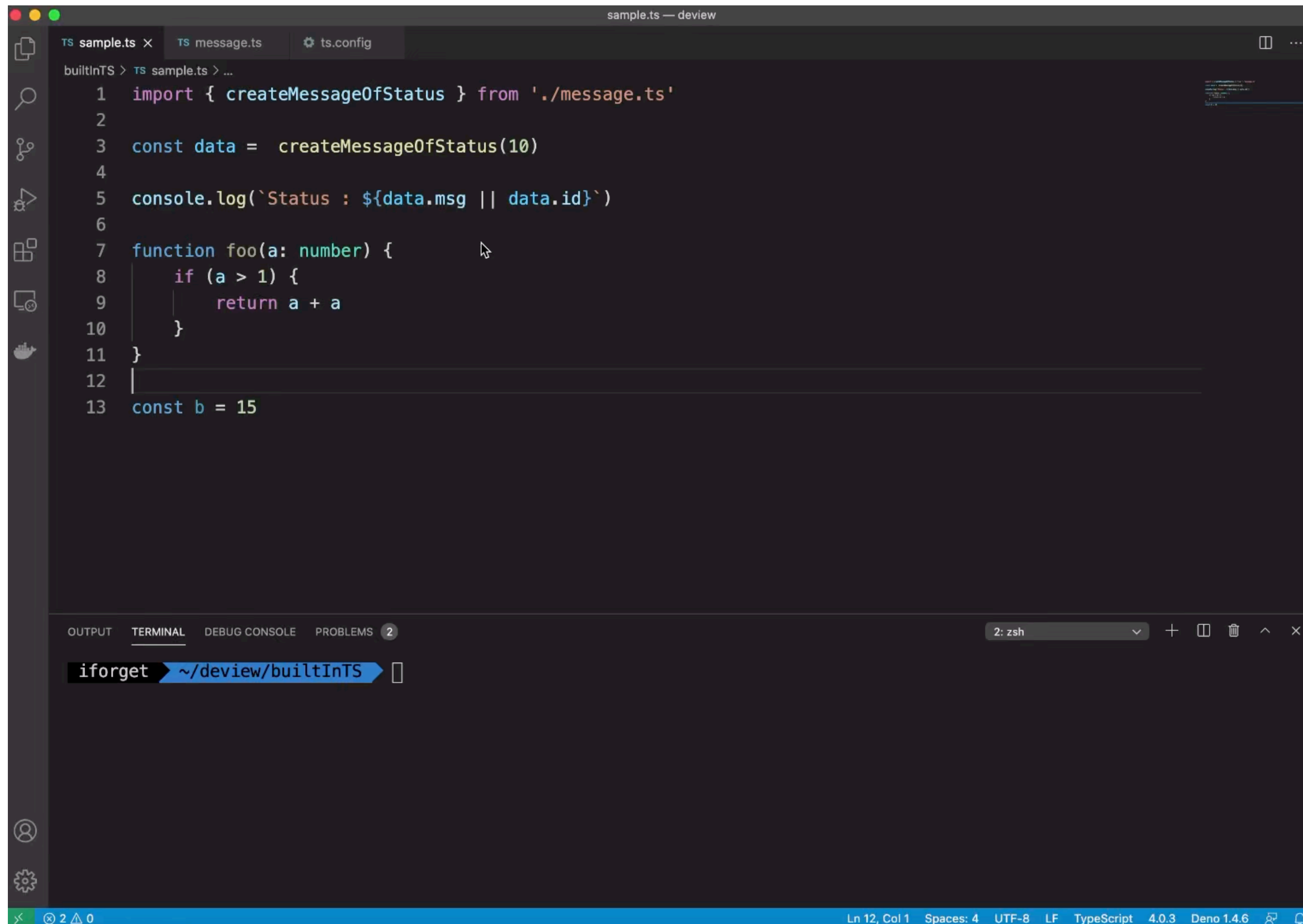


[참조] <https://deno.land/benchmarks>

5. Coding :: Example



Built in TypeScript



```
sample.ts — devview
TS sample.ts x TS message.ts ts.config
builtInTS > TS sample.ts > ...
1 import { createMessageOfStatus } from './message.ts'
2
3 const data = createMessageOfStatus(10)
4
5 console.log(`Status : ${data.msg} || data.id`)
6
7 function foo(a: number) {
8     if (a > 1) {
9         return a + a
10    }
11 }
12
13 const b = 15
```

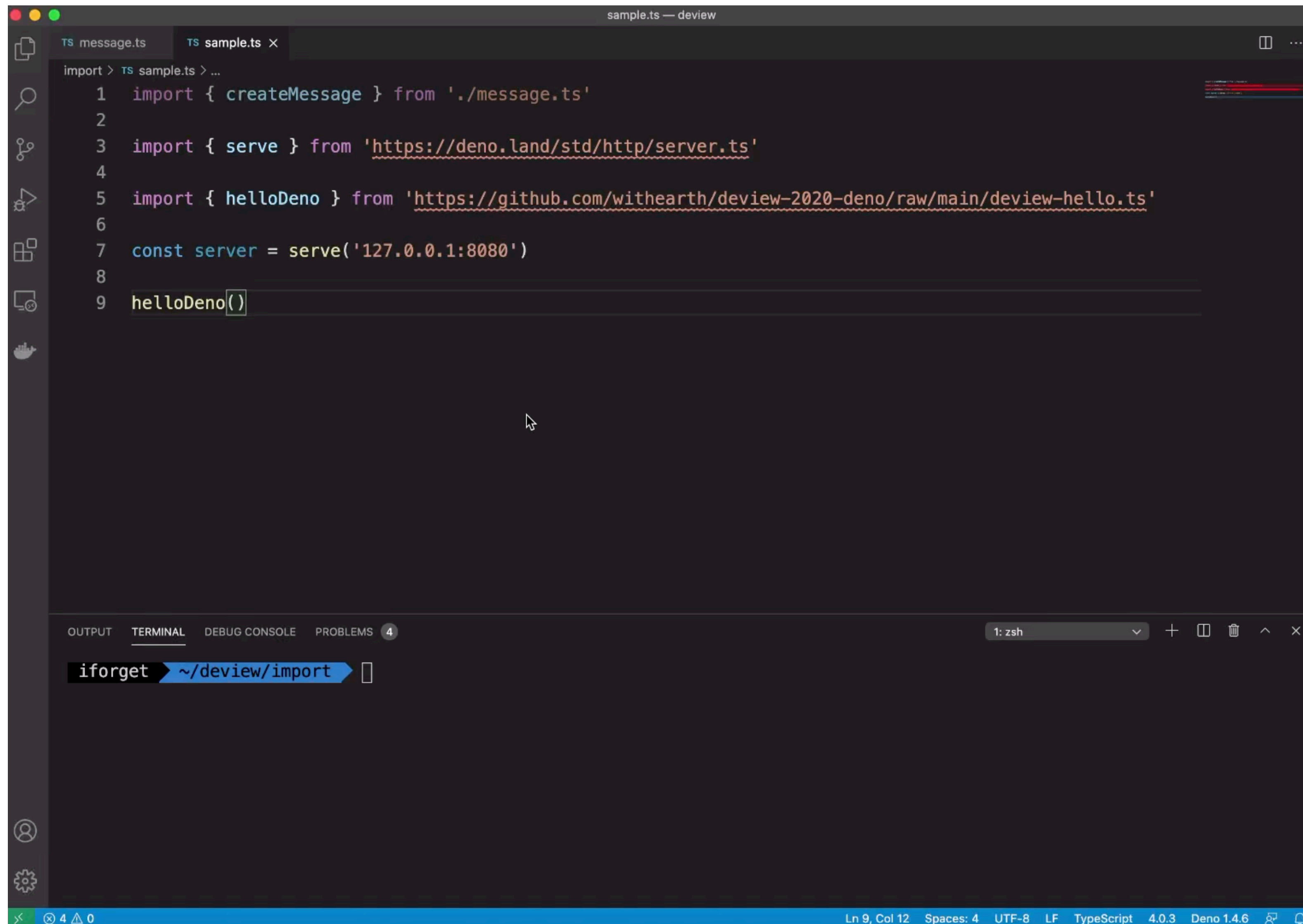
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2

2: zsh

iforget ~/devview/builtInTS

Ln 12, Col 1 Spaces: 4 UTF-8 LF TypeScript 4.0.3 Deno 1.4.6

Import - no package manager



The screenshot shows the DevView IDE interface. The main editor window displays a TypeScript file named `sample.ts` with the following code:

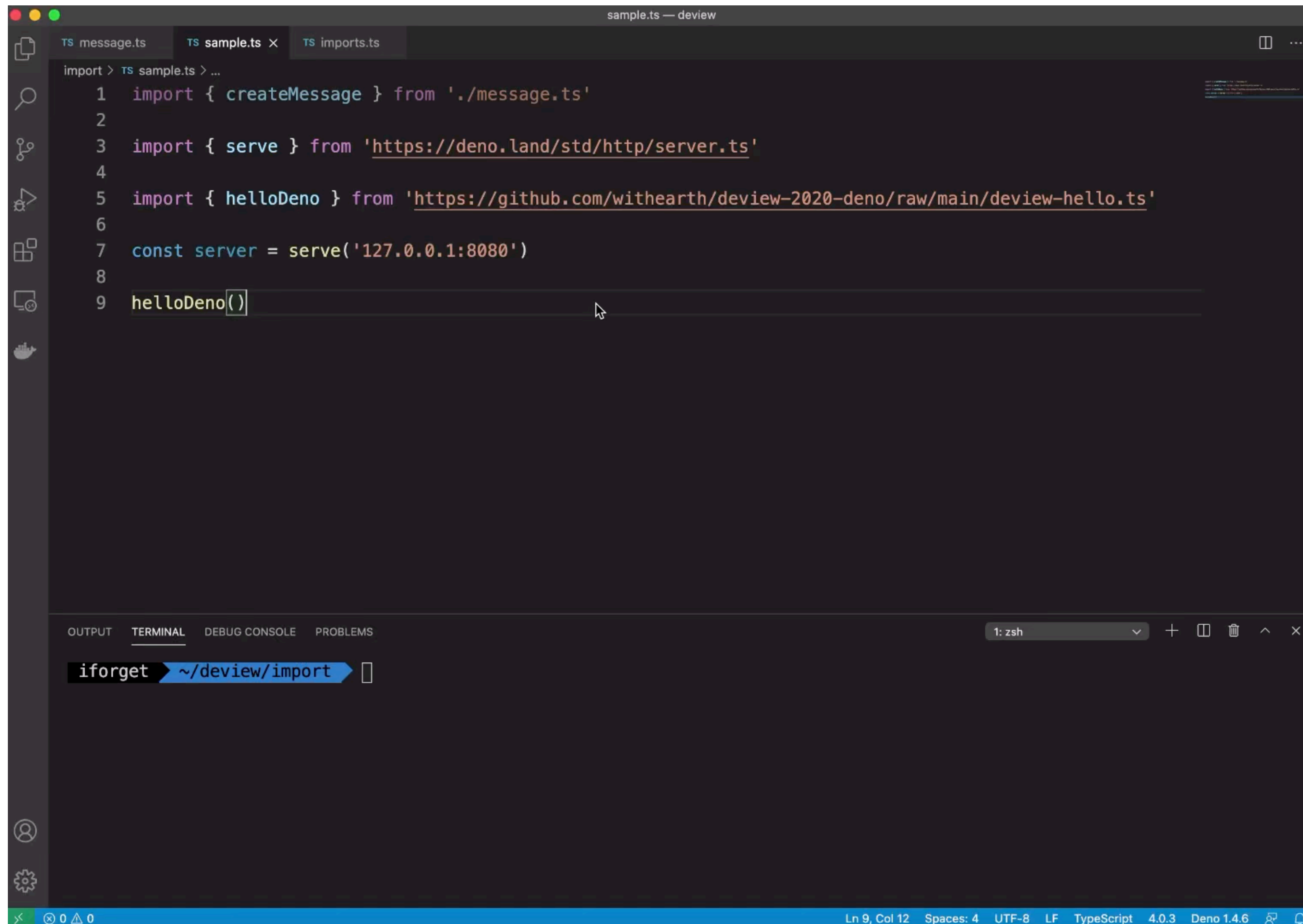
```
import > TS sample.ts > ...
1 import { createMessage } from './message.ts'
2
3 import { serve } from 'https://deno.land/std/http/server.ts'
4
5 import { helloDeno } from 'https://github.com/withearth/devview-2020-deno/raw/main/devview-hello.ts'
6
7 const server = serve('127.0.0.1:8080')
8
9 helloDeno()
```

The bottom panel shows the **TERMINAL** tab with the following prompt:

```
iforget ~/devview/import
```

The status bar at the bottom indicates the current file is `sample.ts` at line 9, column 12, using 4 spaces, UTF-8 encoding, LF line endings, TypeScript 4.0.3, and Deno 1.4.6.

Import - combine import



```
import > TS sample.ts > ...
1  import { createMessage } from './message.ts'
2
3  import { serve } from 'https://deno.land/std/http/server.ts'
4
5  import { helloDeno } from 'https://github.com/withearth/devview-2020-deno/raw/main/devview-hello.ts'
6
7  const server = serve('127.0.0.1:8080')
8
9  helloDeno()
```

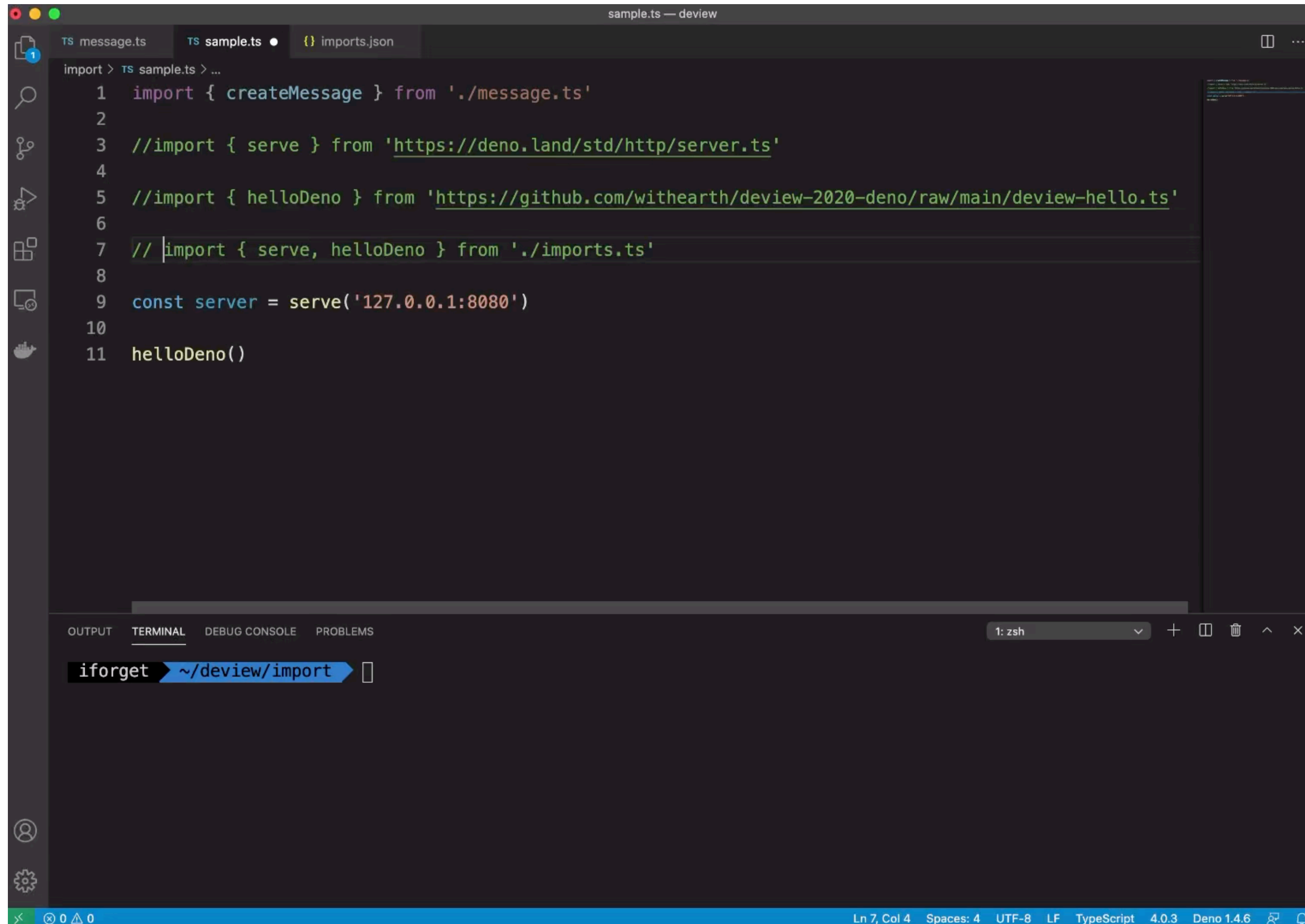
OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS

1: zsh

iforget ~/devview/import

Ln 9, Col 12 Spaces: 4 UTF-8 LF TypeScript 4.0.3 Deno 1.4.6

Import - importMap



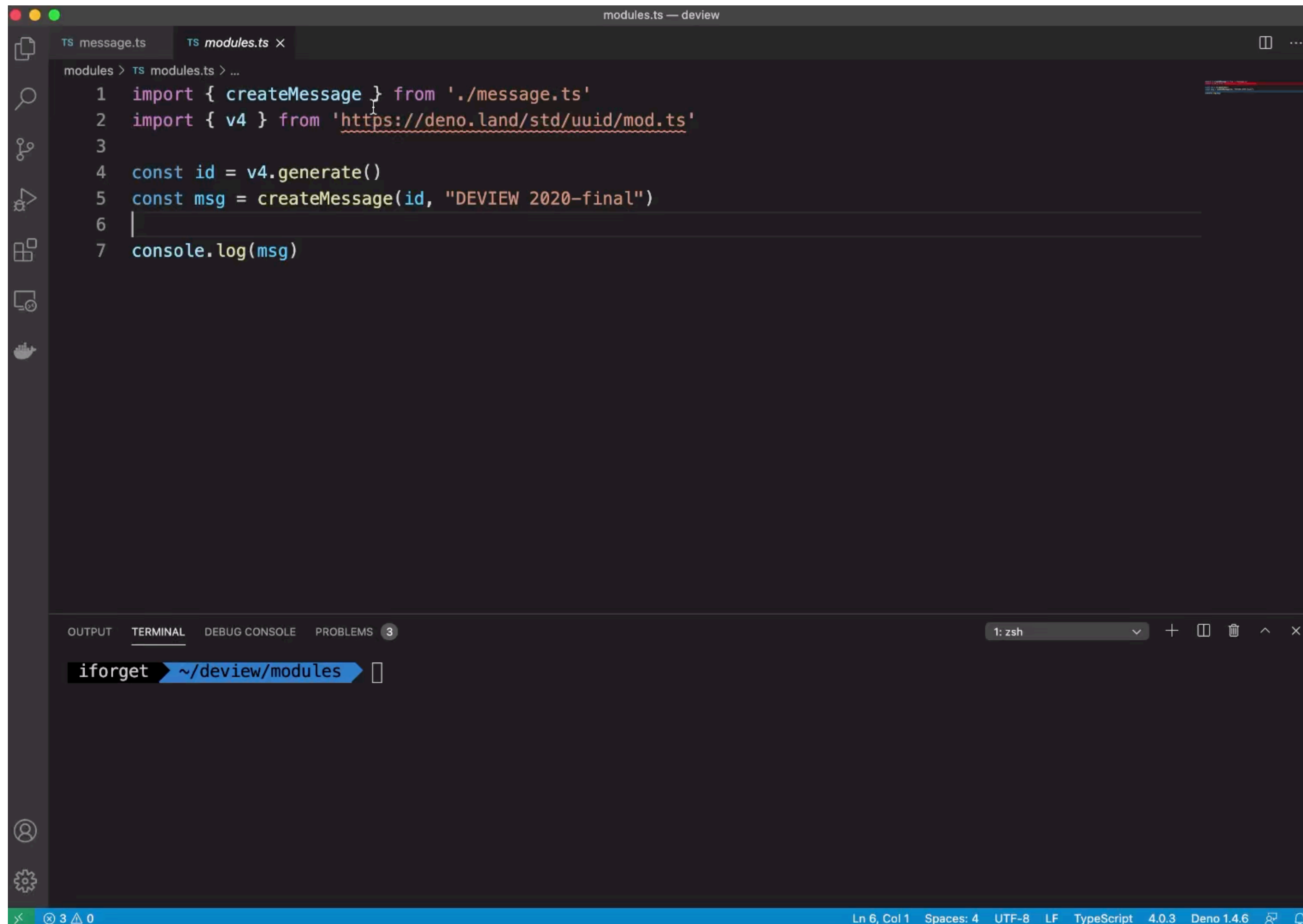
The screenshot shows the DevView IDE interface. The main editor displays a TypeScript file named `sample.ts` with the following code:

```
import > TS sample.ts > ...
1  import { createMessage } from './message.ts'
2
3  //import { serve } from 'https://deno.land/std/http/server.ts'
4
5  //import { helloDeno } from 'https://github.com/withearth/devview-2020-deno/raw/main/devview-hello.ts'
6
7  // import { serve, helloDeno } from './imports.ts'
8
9  const server = serve('127.0.0.1:8080')
10
11 helloDeno()
```

The terminal window at the bottom shows the command prompt with the text `iforget ~/devview/import` and a cursor.

The status bar at the bottom indicates the file is at Line 7, Column 4, using UTF-8 encoding, LF line endings, and is a TypeScript file. The Deno version is 1.4.6.

Modules - node_modules



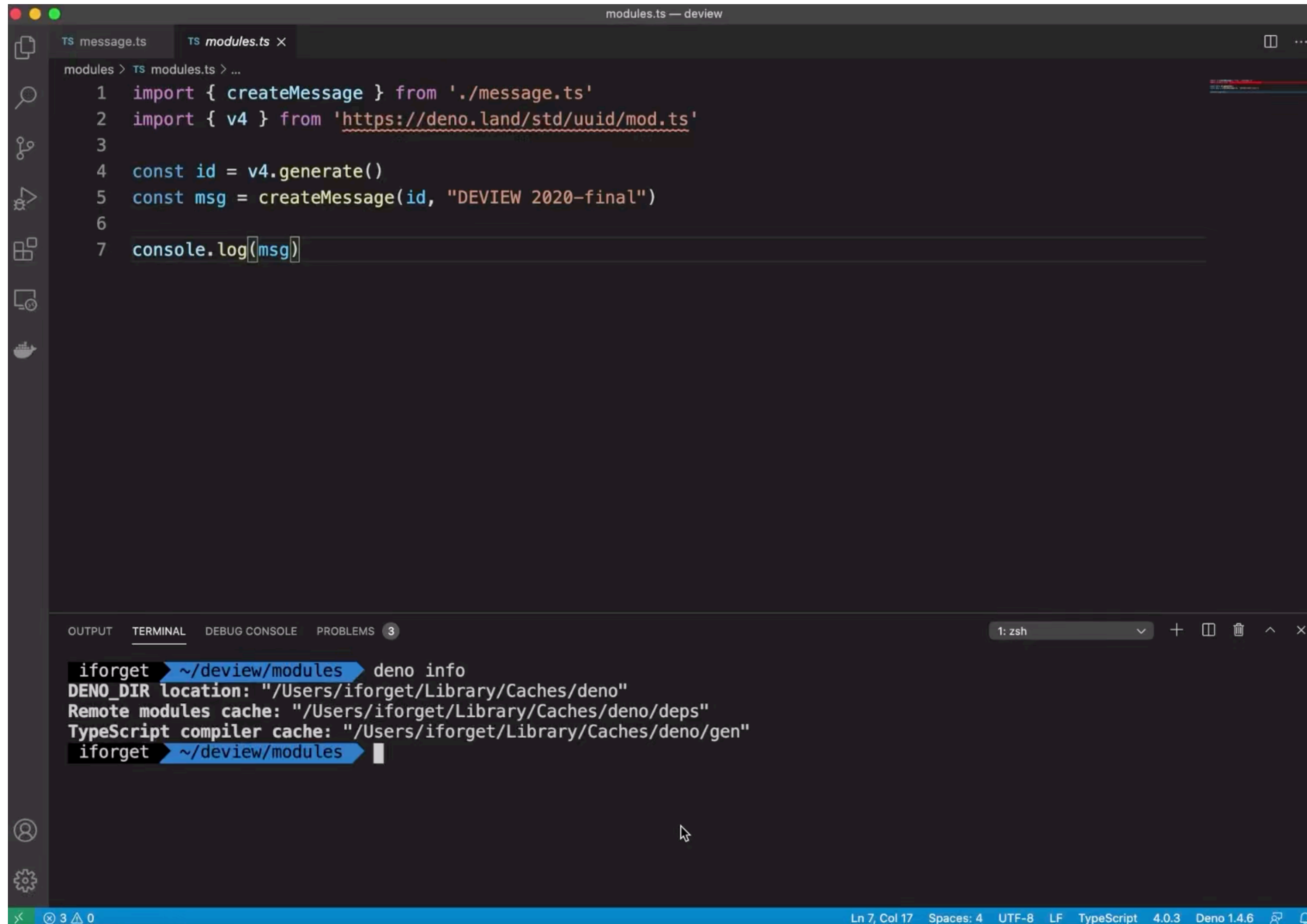
The screenshot shows a VS Code editor window with a file named `modules.ts` open. The file contains the following TypeScript code:

```
1 import { createMessage } from './message.ts'
2 import { v4 } from 'https://deno.land/std/uuid/mod.ts'
3
4 const id = v4.generate()
5 const msg = createMessage(id, "DEVVIEW 2020-final")
6
7 console.log(msg)
```

Below the editor, the **TERMINAL** panel is active, showing a shell prompt `iforget` and the current directory `~/devview/modules`.

The status bar at the bottom indicates the file is at `Ln 6, Col 1`, using `Spaces: 4`, `UTF-8` encoding, `LF` line endings, `TypeScript` language, `4.0.3` version, and `Deno 1.4.6` runtime.

Modules - global cache



The screenshot shows a code editor with a file named `modules.ts` open. The code in the file is as follows:

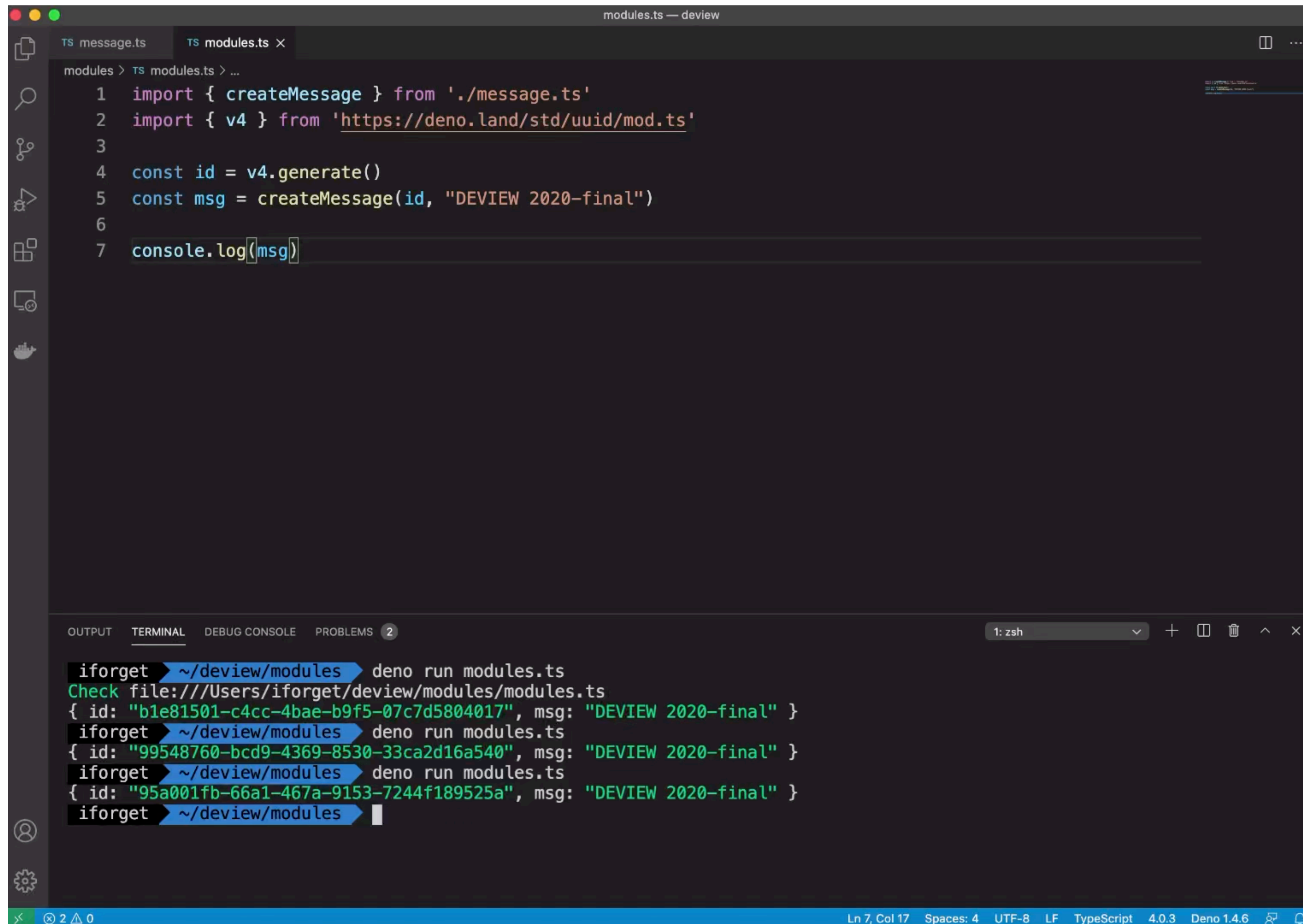
```
1 import { createMessage } from './message.ts'
2 import { v4 } from 'https://deno.land/std/uuid/mod.ts'
3
4 const id = v4.generate()
5 const msg = createMessage(id, "DEVVIEW 2020-final")
6
7 console.log(msg)
```

Below the code editor is a terminal window. The terminal shows the output of the command `deno info` run in the directory `~/devview/modules`:

```
iforget ~/devview/modules deno info
DENO_DIR location: "/Users/iforget/Library/Caches/deno"
Remote modules cache: "/Users/iforget/Library/Caches/deno/deps"
TypeScript compiler cache: "/Users/iforget/Library/Caches/deno/gen"
iforget ~/devview/modules
```

The terminal window also shows the command prompt `1: zsh` and a status bar at the bottom indicating the file is at `Ln 7, Col 17` with `Spaces: 4`, `UTF-8` encoding, `LF` line endings, `TypeScript 4.0.3`, and `Deno 1.4.6`.

Modules - reload & watch



The screenshot shows a VS Code editor window with a file named 'modules.ts' open. The file contains the following TypeScript code:

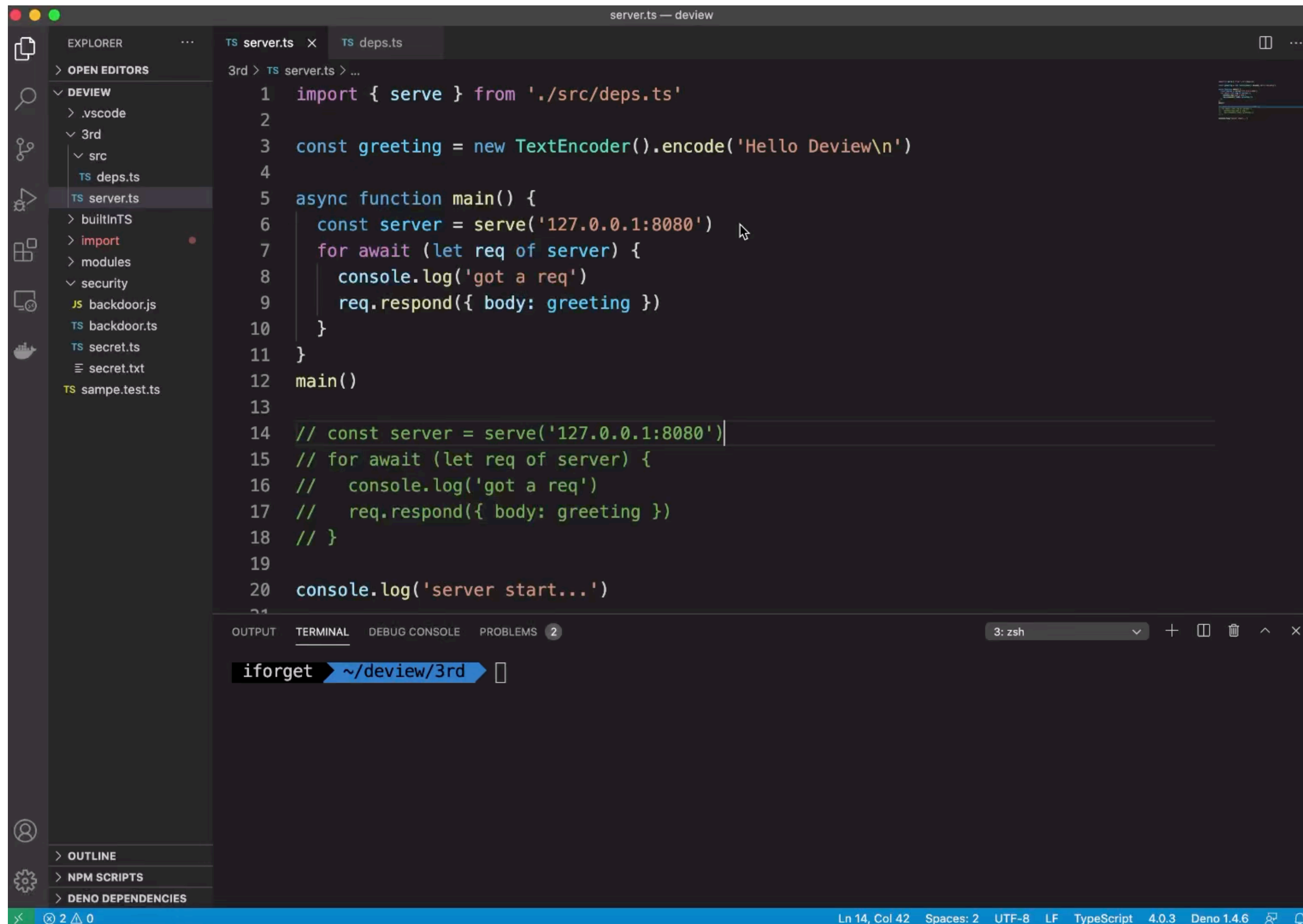
```
1 import { createMessage } from './message.ts'
2 import { v4 } from 'https://deno.land/std/uuid/mod.ts'
3
4 const id = v4.generate()
5 const msg = createMessage(id, "DEVVIEW 2020-final")
6
7 console.log(msg)
```

Below the editor, the 'TERMINAL' tab is active, showing the command 'deno run modules.ts' being executed multiple times. The output of each command is a JSON object containing a UUID and the message 'DEVVIEW 2020-final'.

```
iforget ~/devview/modules deno run modules.ts
Check file:///Users/iforget/devview/modules/modules.ts
{ id: "b1e81501-c4cc-4bae-b9f5-07c7d5804017", msg: "DEVVIEW 2020-final" }
iforget ~/devview/modules deno run modules.ts
{ id: "99548760-bcd9-4369-8530-33ca2d16a540", msg: "DEVVIEW 2020-final" }
iforget ~/devview/modules deno run modules.ts
{ id: "95a001fb-66a1-467a-9153-7244f189525a", msg: "DEVVIEW 2020-final" }
iforget ~/devview/modules
```

The status bar at the bottom indicates the file is at Line 7, Column 17, using UTF-8 encoding, LF line endings, and TypeScript 4.0.3. The Deno version is 1.4.6.

Top Level/First class await & lock.json

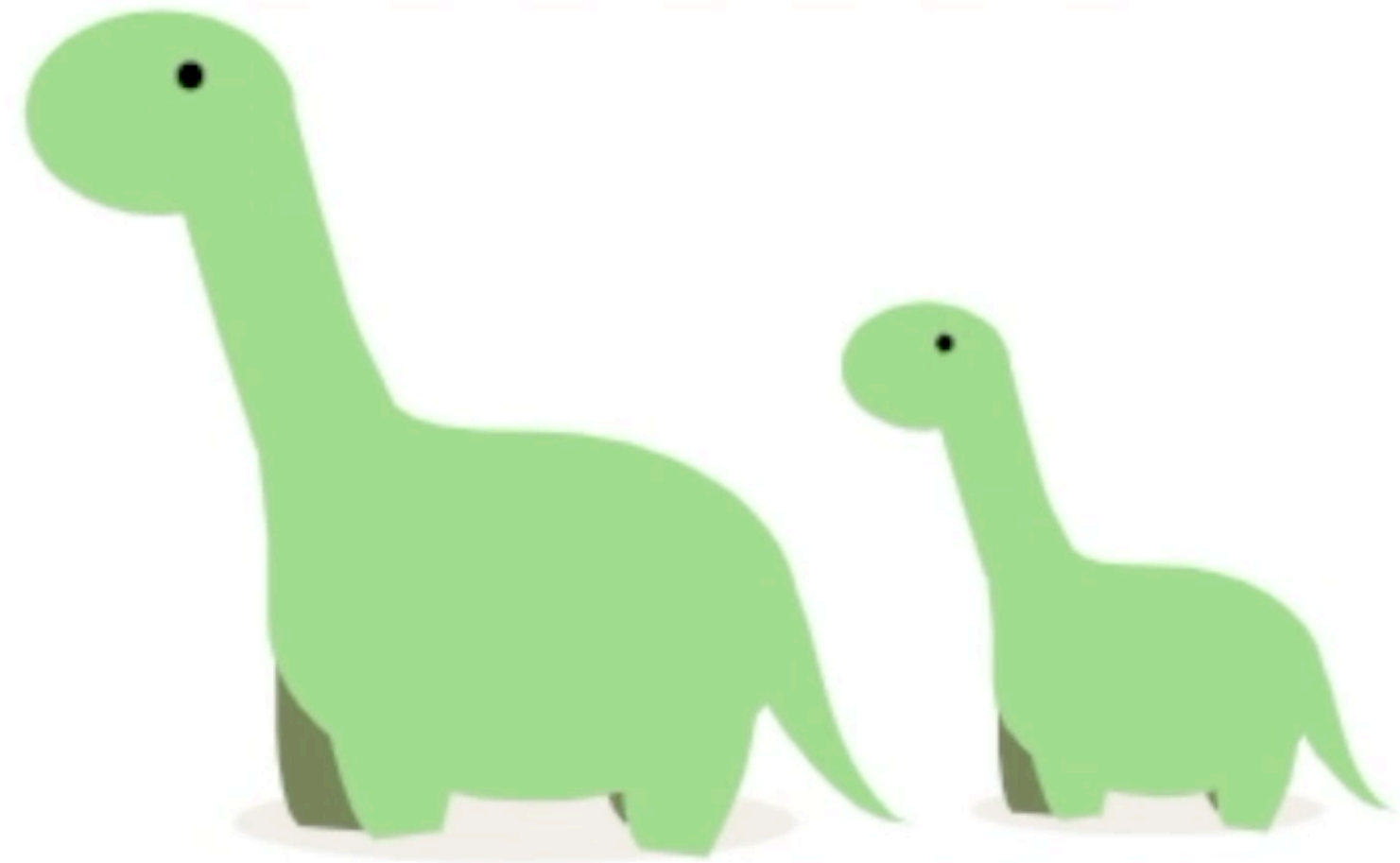


```
server.ts — devview
EXPLORER
> OPEN EDITORS
DEVIEW
> .vscode
> 3rd
  > src
    TS deps.ts
    TS server.ts
  > builtinTS
  > import
  > modules
  > security
    JS backdoor.js
    TS backdoor.ts
    TS secret.ts
    secret.txt
    TS sampe.test.ts
  > OUTLINE
  > NPM SCRIPTS
  > DENO DEPENDENCIES

3rd > TS server.ts > ...
1  import { serve } from './src/deps.ts'
2
3  const greeting = new TextEncoder().encode('Hello Devview\n')
4
5  async function main() {
6    const server = serve('127.0.0.1:8080')
7    for await (let req of server) {
8      console.log('got a req')
9      req.respond({ body: greeting })
10   }
11 }
12 main()
13
14 // const server = serve('127.0.0.1:8080')
15 // for await (let req of server) {
16 //   console.log('got a req')
17 //   req.respond({ body: greeting })
18 // }
19
20 console.log('server start...')
21

OUTPUT TERMINAL DEBUG CONSOLE PROBLEMS 2
3: zsh
iforget ~/devview/3rd
```

6. Preparing for a future with _____.



#리액트

웹팩 없는 세상에서 살고싶다

too many configs

|

불만의 품격 SPA 가 아닌 서비스에 React.js 이점 있나?

민

type을 IDE가 잘좀 추천해주면 좋겠다

그냥+쉬고+싶다.

#TRADE OFF

필요한게 하나인데, 모든게 필요하네요

세상에 무시해도 되는 불편함은 없다

async/await 계속 붙이는 거 안불편해?

여러군데 쓰이는 공통타입에 any를 쓰기 껄끄러운데. #TS/JS

ECMA20XX 피로해

이런이런 |오우글록 크립트구머노

혼자급한데



디노가 나온지 꽤 되었네요. 안정화도 되었고, 널리 알려지기도 했고. 한국에선 Devview2020 에서 발표도

deno 관해 후회하는 10가지 - Ryan Dahl - JSConf KOREA

조회수 472,309회 • 2035. 8. 6.

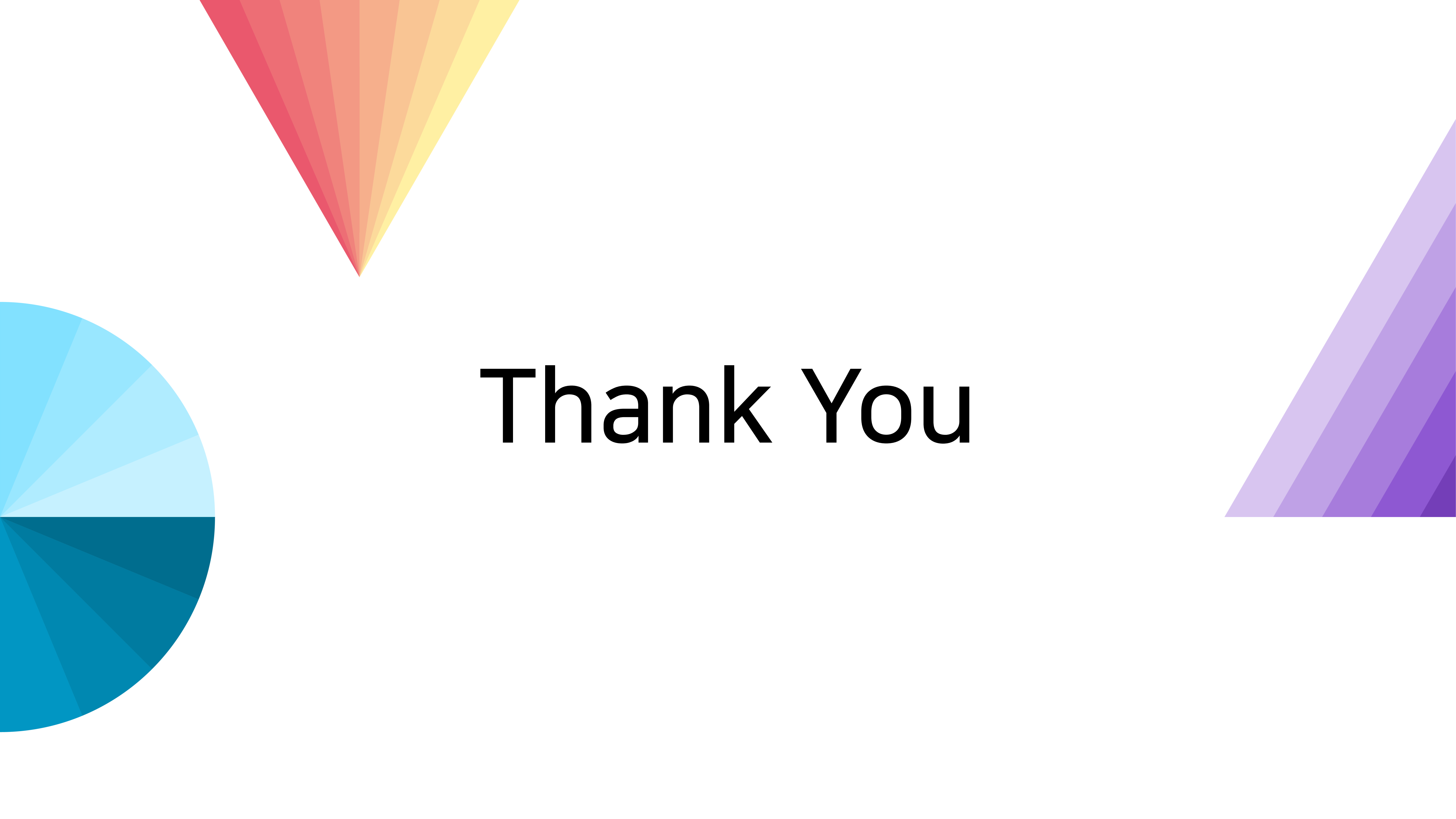
👍 좋아요 💬 싫어요 ➦ 공유 📌 저장 ...

Preparing for a future with 불편함.



Q & A

keunbae.bae@navercorp.com



Thank You