



Q&A (& Lunch)  
Come talk with us!

Capri 2 @ 12:30



Chat 1:1 with a GitHub  
Expert on Day 2



0123456789@BCDEFCHIJKLMNOPQRSTUVWXYZ.,!,-@%<~>+\_,?C|S>:=;"?"=!<,.



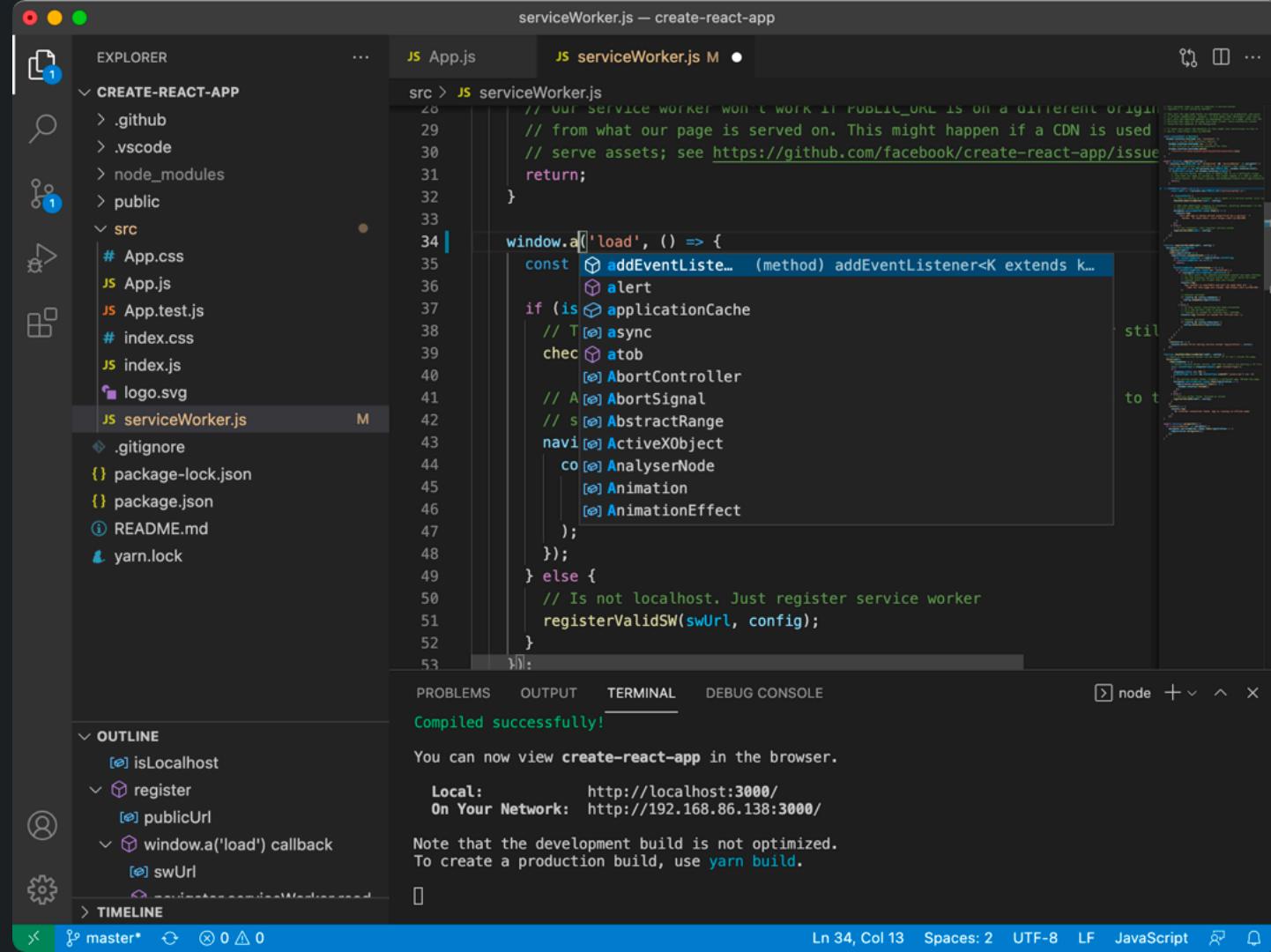




```
share_test.go + ~/global-marketing-solutions/happy-marketeer/backend/aws/lambdas/share - VIM
>>> projectStatus = "In Progress"
)$
$
func init() {
myDynamo.ProjectDelete(email, projectName)
}
$
func TestUpdateItemWithCopy(t *testing.T) {
myDynamo.AddProject(email, projectName, startDate, dueDate)
}

>>> updateItemWithCopy
$ updateItemInput
$ updateItem      tProject(emailOfSharee, projectName)
$ updateItemWithCopy
$ updateItemWithCopy      th another owner: %#v\n", item)
$
if *item["sk"].S != pure.BuildSortKey(emailOfSharee, projectName) {
t.Fatalf("sk: %v\n", item["sk"]))
}
if *item["startDate"].S != startDate {
t.Fatalf("startDate actual: %v\n", item["startDate"]))
}
if *item["dueDate"].S != dueDate {
t.Fatalf("dueDate actual: %v\n", item["dueDate"]))
}
if *item["projectStatus"].S != projectStatus {
t.Fatalf("projectStatus actual: %v\n", item["projectStatus"]))
}
myDynamo.ProjectDelete(email, projectName)
16
```

```
share test.go [+]  
share test.go|29 col 2 error| undefined: updateItmWithCopy$
```



👤 How will we move past using chat for every application of AI?

🤖 Sorry, I'm just a chatbot! I'm afraid I don't have a good answer for that.

I



Reaching for  
**AI-Native**  
**User Experiences**

Idan Gazit  
Sr. Director of Research  
GitHub Next



twitter: @githubnext  
github: githubnext.com



twitter: @idangazit  
email: idan@babka.social



ghost text



chat

doing



planning



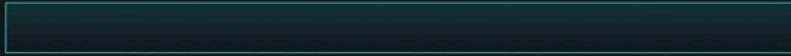
## Ghost Text

“Guess what I want”

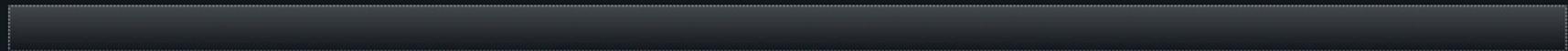
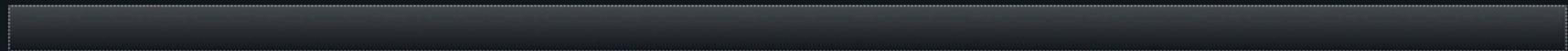
## Chat

“Discuss what I want”

line function



*more code...*



pull request

## The AI Slot Machine



— Request —  
← Response ←

task -----> code





JavaScript    HTML    CSS    Java    Python    C++    PHP    F#

```
1 // Import required module
2 const process = require('process');
3
4 // Define a class for Terminal
5 class Terminal_Dimensions {
6   constructor() {
7     this.width = process.stdout.columns;
8     this.height = process.stdout.rows;
9   }
10 }
11
12 // Implement timing
13 let timing_start;
14 const time_limit = 9999; // maximum time limit in milliseconds
15
16 const timing_start_m_ = () => {
17   timing_start = new Date();
18 };
```

Font size

Weight

Width

Slant

Texture healing

Ligatures

Grid

# Add a Rust example to the playground #59

[Edit](#)[New issue](#)[Open](#)

idan opened on May 21

edited by idan · Edits · ...

The code playground has tabs which show examples for a number of languages. Add a tab with some example code in Rust.

Acceptance criteria:

- Add a Rust code sample to the code samples
- Add a dependency on the `@codemirror/lang-rust` package from NPM at version 6.0.1
- Add a tab in the font playground which displays the rust sample using the codemirror package for rust syntax highlighting



idan changed the title ~~Add a ruby example to the playground~~ Add a Rust example to the playground on May 21



dsyme mentioned this on Jun 4

**Assignees**

No one - Assign yourself

**Labels**

No labels

**Projects**

No projects

**Milestone**

No milestone

**Development**[Open in Workspace](#)

Create a branch for this issue or link a pull request.

Copilot Workspace Preview Give feedback

copilot-workspace.githubnext.com

githubnext / monospace-site #59 main

Specification ⓘ

Is there a Rust code sample added to the code samples with a tab in the font playground displaying it using the codemirror package for rust syntax highlighting?

Current ⓘ

- No, there is no Rust code sample added to the code samples.
- No dependency on `@codemirror/lang-rust` package in `package.json`.
- No tab for Rust in the font playground in `src/components/font-playground.tsx`.
- Existing code samples include JavaScript, HTML, CSS, Java, Python, C++, PHP, and F# in `src/code-samples.ts`.
- The `src/components/font-playground.tsx` file handles the display of code samples using the CodeMirror package for syntax highlighting.

+ Add item

Proposed ⓘ

- Yes, there is now a Rust code sample added to the code samples in `src/code-samples.ts`.
- The Rust code sample includes a simple "Hello, World!" program.
- Yes, there is now a dependency on `@codemirror/lang-rust` package in `package.json`.
- The `@codemirror/lang-rust` package is added under dependencies with version 6.0.1.
- Yes, there is now a tab for Rust in the font playground in `src/components/font-playground.tsx`.
- The Rust tab displays the Rust code sample using the CodeMirror package for Rust syntax highlighting.

+ Add item

0 New session

Create pull request

This screenshot shows a GitHub Copilot workspace interface. At the top, there's a navigation bar with icons for back, forward, and search, followed by the URL 'copilot-workspace.githubnext.com'. Below the URL is a header with 'Copilot Workspace' and 'Preview' buttons, and a 'Give feedback' link. On the right side of the header are icons for notifications (0), creating a new session, and user profile. The main area shows a 'Specification' card with the title 'Is there a Rust code sample added to the code samples with a tab in the font playground displaying it using the codemirror package for rust syntax highlighting?'. The card is divided into 'Current' and 'Proposed' sections. A pink arrow points to the 'Current' section, which lists five bullet points about the absence of Rust support. Another pink arrow points to the 'Proposed' section, which lists six bullet points about the addition of Rust support. Both sections have a '+ Add item' button at the bottom. The top of the page also shows a repository path 'githubnext / monospace-site #59 main' and various GitHub navigation links like 'Create pull request'.



0 New session



Preview

Give feedback

githubnext / monaspace-site #59 main



Plan

 package.json 

- Add a dependency on `@codemirror/lang-rust` package at version 6.0.1 under dependencies
- + Add item

 src/code-samples.ts 

- Add a Rust code sample to the `codeSamples` array with name "Rust", language "rust", and a simple "Hello, World!" program
- + Add item

 src/components/font-playground.tsx 

- Import `rust` from `@codemirror/lang-rust`
- Add `rust` to the `languages` object with key "rust"
- Add a tab for Rust in the `Tabs.List` and `Tabs.Content` components
- + Add item

Add file to plan

Add general notes

View references



0 New session



## Copilot Workspace

Preview

Give feedback

githubnext / monospace-site #59 main

package.json ...

- ✓ Add a dependency on `@codemirror/lang-rust` package at version 6.0.1 under dependencies

+ Add item

src/code-samples.ts ...

- ✓ Add a Rust code sample to the `codeSamples` array with name "Rust", language "rust", and a simple "Hello, World!" program

+ Add item

src/components/font-playground.tsx ...

- ✓ Import `rust` from `@codemirror/lang-rust`
- ✓ Add `rust` to the `languages` object with key "rust"
- ✓ Add a tab for Rust in the `Tabs.List` and `Tabs.Content` components

+ Add item

```
2 import { FontKey, axes, fonts } from "@for
3 import { cpp } from "@codemirror/lang-cpp";
4 import { css } from "@codemirror/lang-css";
5 import { html } from "@codemirror/lang-html"
6 import { java } from "@codemirror/lang-java"
7 import { javascript } from "@codemirror/lar
8 import { php } from "@codemirror/lang-php";
9 import { python } from "@codemirror/lang-py
10 import { StreamLanguage } from "@codemirror/]
11 import { fSharp } from "@codemirror/legacy-m
12 import {
13   Checkbox,
14   FormControl,
15   Grid,
16   Select,
17   Stack,
18   Text,
19 } from "@primer/react-brand";
20 import * as Tabs from "@radix-ui/react-tabs
21 import {
22   atomone,
23   darcula,
24   defaultSettingsAtomone,
25   defaultSettingsDarcula,
26   defaultSettingsDracula,
27   defaultSettingsGithubDark,
28   defaultSettingsMaterial
29 }
```

Revise plan... >

Where is the code I need to edit?  
How does it work right now?  
What do I need to change about it?

Writing Code

---

*Most of the time*

---

*A lot less time*



githubnext.com/projects/copilot-workspace

doing



planning

## Ghost Text

“Guess what I want”

## Structured Exchange

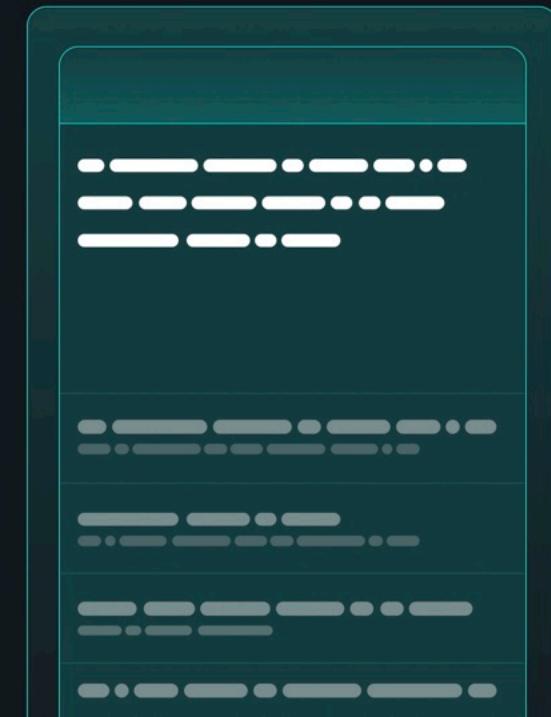
“Here’s what I want”

## Chat

“Discuss what I want”



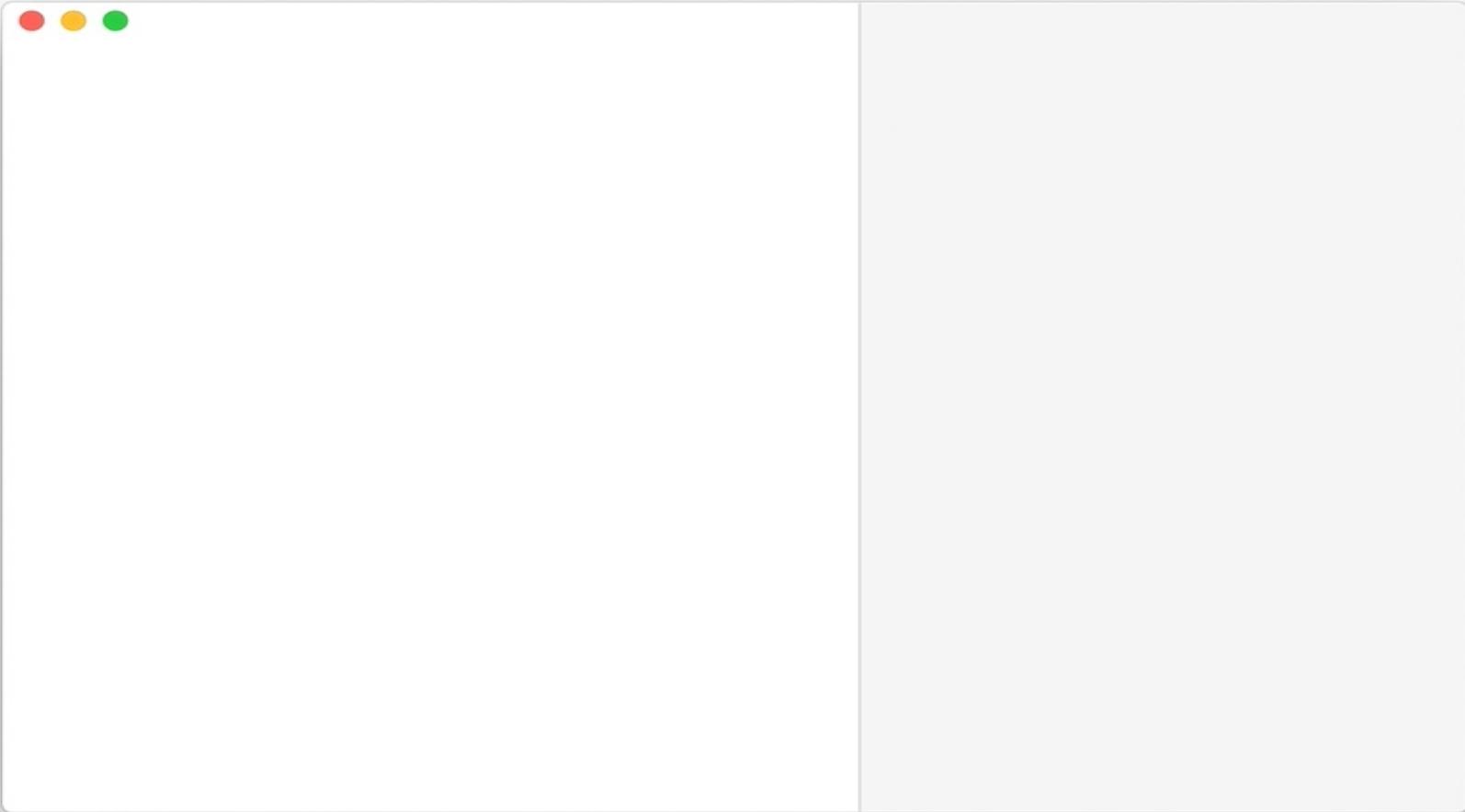
[githubnext.com/projects/speclang](https://githubnext.com/projects/speclang)



When was the last time that you

# wrote assembler?

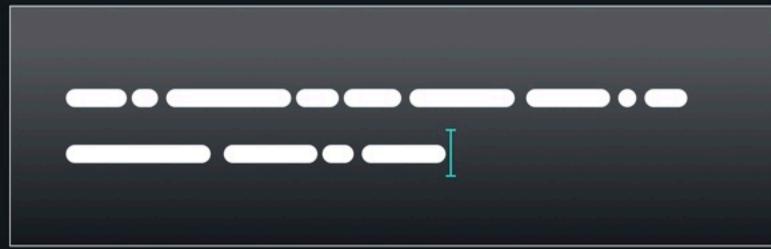
```
mov al, [num1]
add al, [num2]
mov [result], al
mov eax, 1
mov ebx, 0
int 0x80
```



## Bicameral UX



Natural Language



Formal Syntax

# CONFIGURATION

The `run` function is the entry point of the script when it's run by the command line (that is, it's not being used as a library). The `docco` command in `bin` simply runs `require('../docco.js').run()`.

The package commander will be used to parse command-line parameters. Commander will generate a configuration object based on how it's configured using its `option` method.

For example the line:

```
.option('-L, --languages [file]', 'use a custom languages.
```

Means that if Docco Next is run with `--languages`

```
./some/file.json, the config object will include { languages:  
"./some/file.json"}.
```

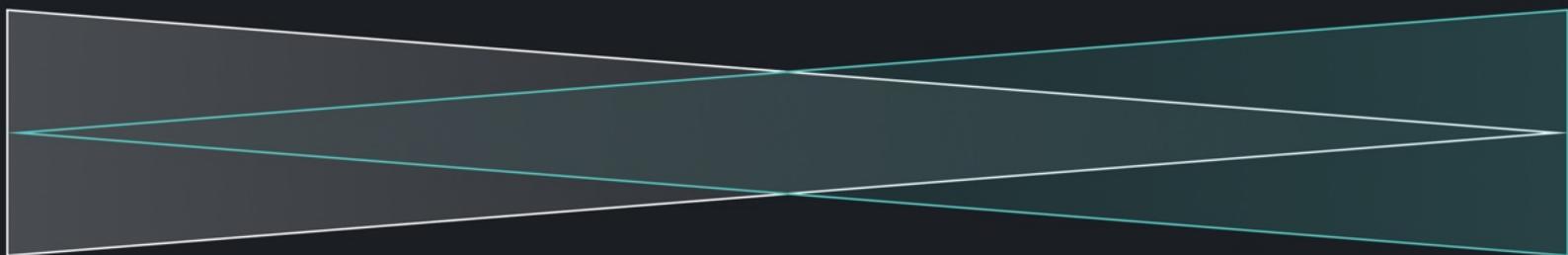
Commander also provides the handy `helpInformation()` method, which will print out how the command is used.

This is the full set of options available in Docco Next:

```
121  async function run (args = process.argv) {  
122    commander  
123      .name('docco')  
124      .version(version)  
125      .usage('[options] files')  
126      .option('-L, --languages [file]', 'use a custom languages.json')  
127      .option(  
128        '-l, --layout [name]',  
129        'choose a layout (default, parallel or classic)'  
130      )  
131      .option('-s, --shiki-theme [shikiTheme]', 'choose a shiki theme')  
132      .option('-o, --output [path]', 'output to a given folder')  
133      .option('-c, --css [file]', 'use a custom css file')  
134      .option('-p, --plugin [file]', 'use a custom plugin file')  
135      .option('-t, --template [file]', 'use a custom .ejs template')  
136      .option(  
137        '-e, --inputExtension [ext]',  
138        'assume a file extension for all inputs'  
139      )  
140      .option('-m, --marked [file]', 'use custom marked options')  
141      .option(  
142        '-x, --outputExtension [ext]',  
143        'set default file extension for all outputs'  
144      )  
145      .parse(args)  
146    if (commander.args.length) {  
147      const config = { ...commander.opts(), args: commander.args }  
148      await cmdLineNormalise(config)  
149      configure(config)  
150      await cmdLineSanityCheck(config)  
151  
152      await documentAll(config)  
153    } else {
```

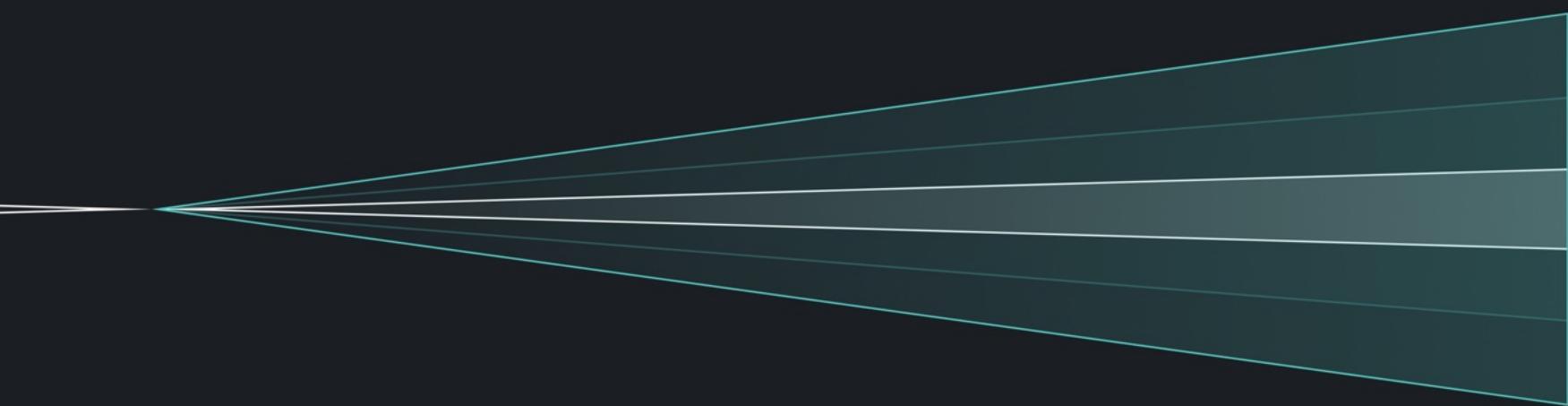
Tools

People



Tools

People



```
> end of file  
> for more, visit:  
> githubnext.com
```

# Thank you!



twitter icon @githubnext  
link icon [githubnext.com](https://githubnext.com)



twitter icon @idangazit  
email icon [idan@babka.social](mailto:idan@babka.social)