

Make Angular CLI Faster

Ng Conf Colombia

15 de agosto de 2020

Make Angular CLI Faster

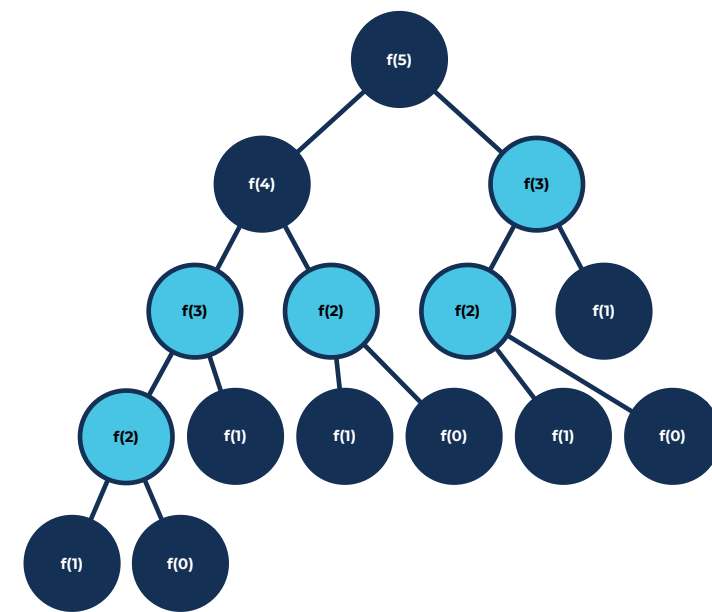
With Nx!

@jeffbrcross

Principal Architect

jeff@nrwl.io





Memoization

What is it?



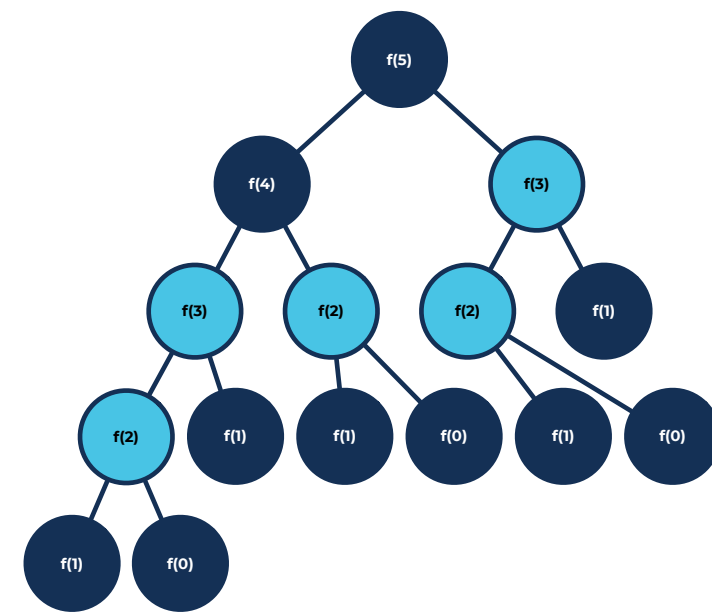
Nx's Computation Memoization

How Nx memoizes computations



Make Angular CLI Faster

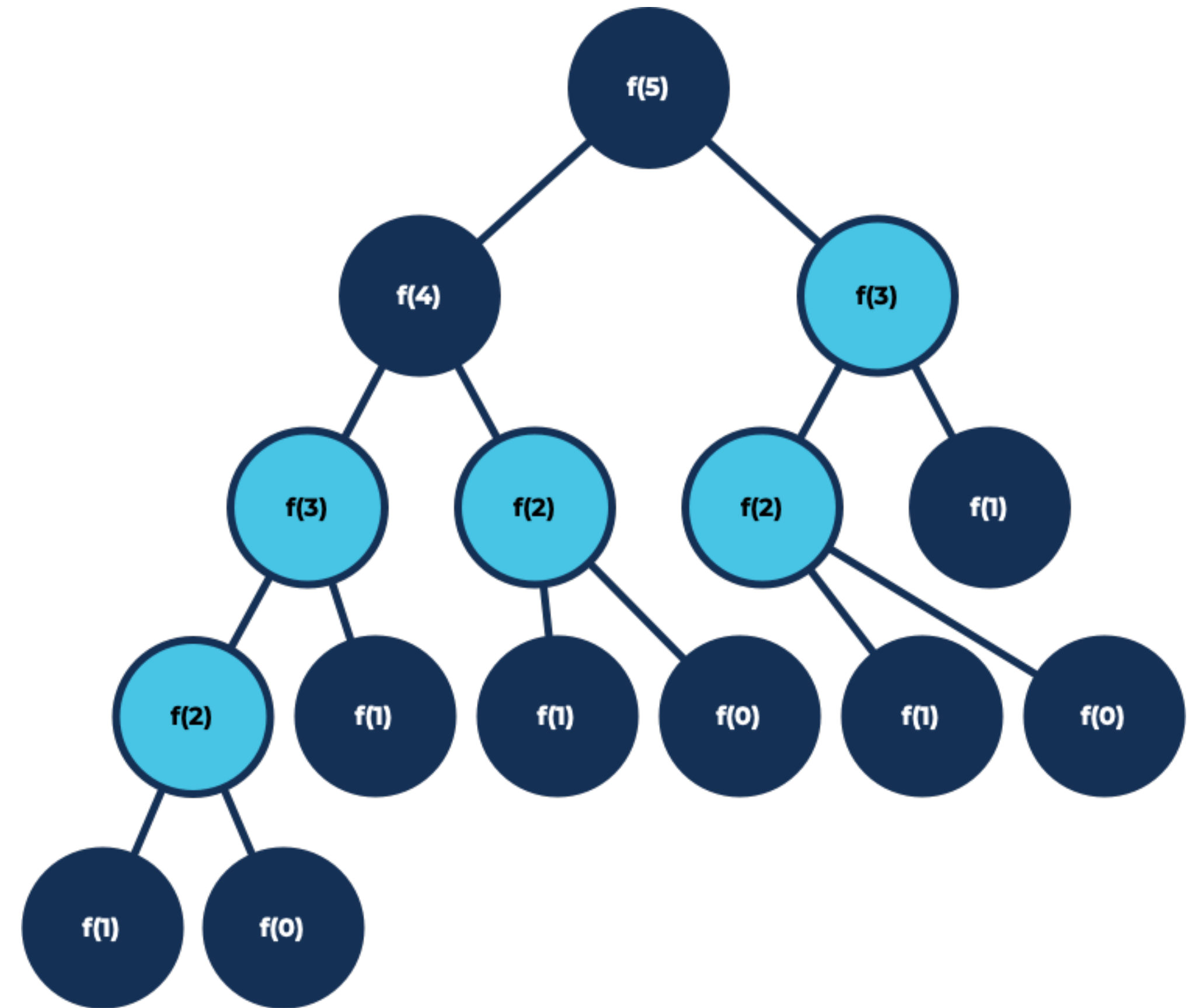
How to get faster builds with Angular CLI by using computation memoization

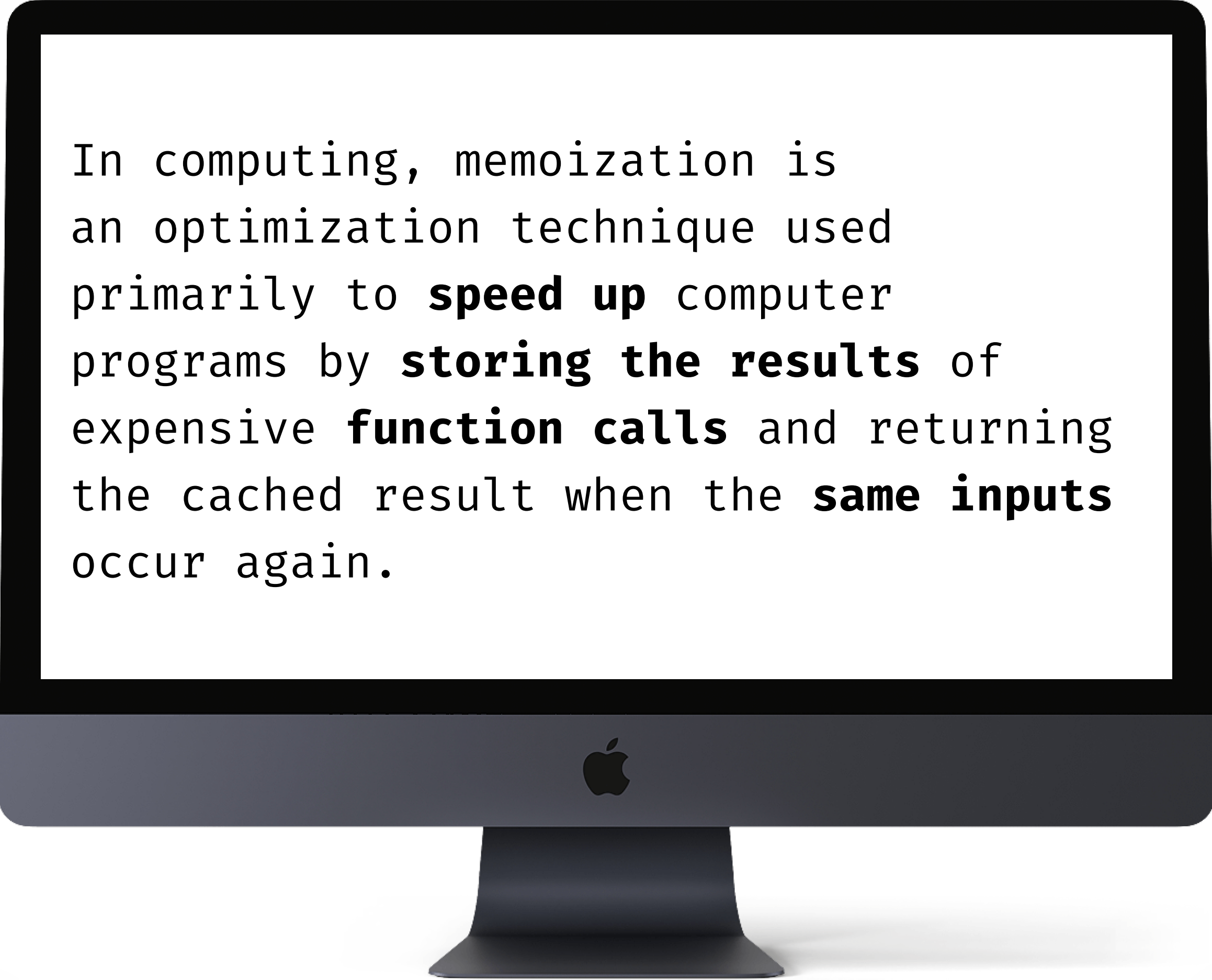


Memoization

What is it?

Memoization

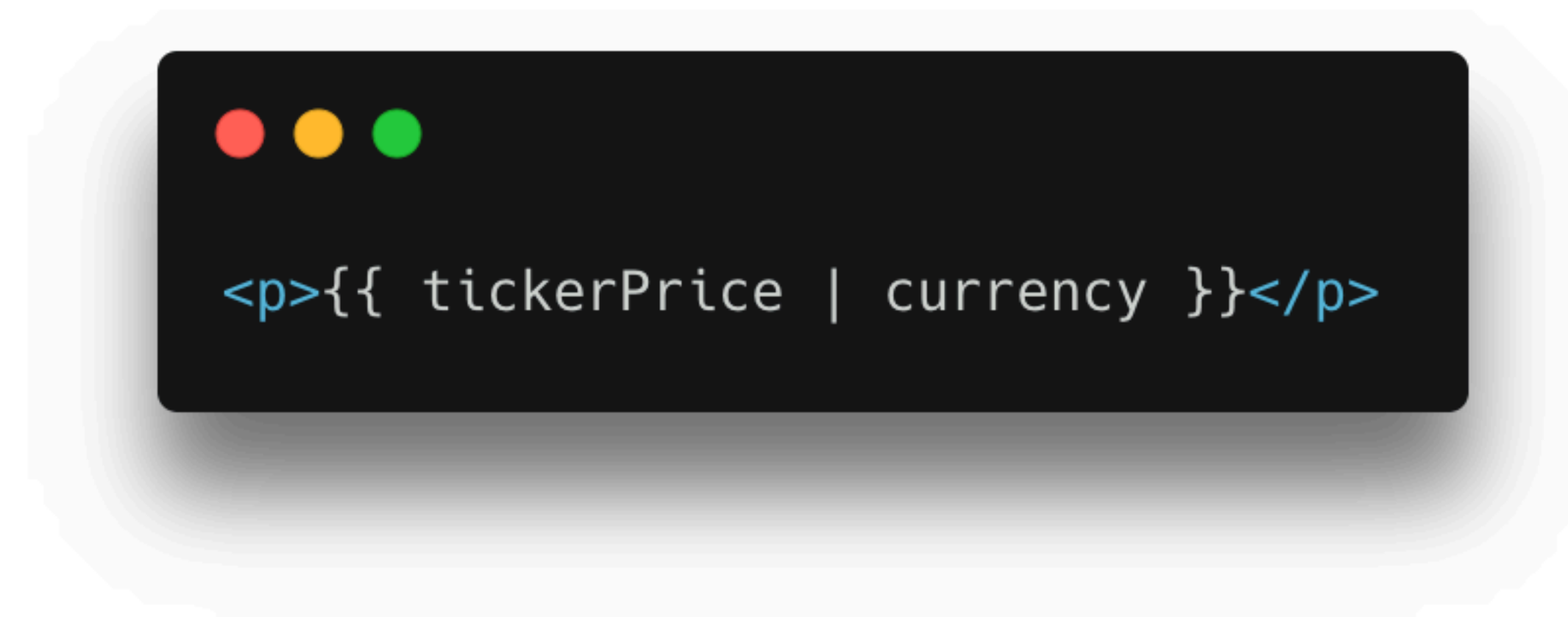


A dark grey computer monitor with a black bezel and a stand. The screen is white and displays text in a monospaced font. The text explains the concept of memoization in computing, highlighting key terms like 'speed up', 'storing the results', 'function calls', and 'same inputs' in bold.

In computing, memoization is an optimization technique used primarily to **speed up** computer programs by **storing the results** of expensive **function calls** and returning the cached result when the **same inputs** occur again.

Angular Pipes use Memoization

When an Angular Pipe is “pure”, it will **cache pipe outputs** and replay the results if the pipe is called with the **same input** again.



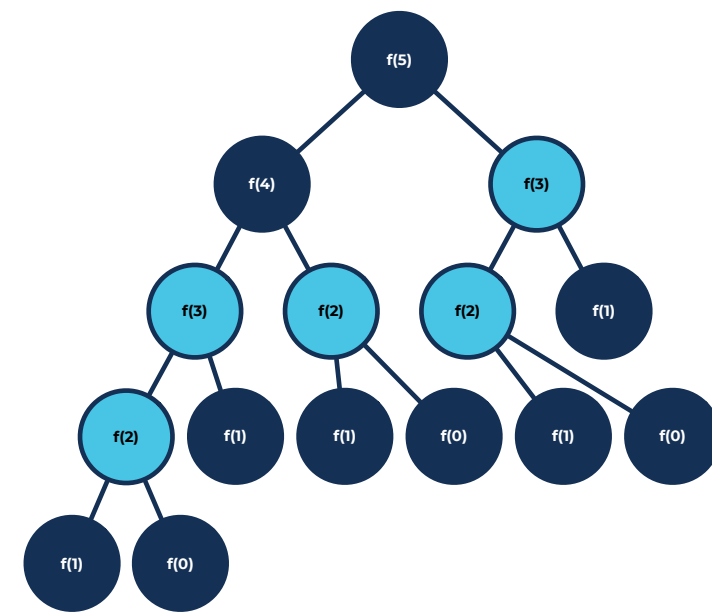
```
this.tickerPrice = 5; // currency(5) ⇒ '$5.00'  
this.tickerPrice = 7.5; // currency(7.5) ⇒ '$7.50'  
this.tickerPrice = 5; // _cache[5] ⇒ '$5.00'
```

Internal Memoization Cache

Input	Result
5	\$5.00
7.5	\$7.50



What else can we memoize?



Memoization

What is it?



Nx's Computation Memoization

How Nx memoizes computations



Make Angular CLI Faster

How to get faster builds with Angular CLI by using computation memoization



Nx's Computation Memoization

How Nx memoizes computations

Make Angular CLI Faster

Ng Conf Colombia

15 de agosto de 2020

Nx's Computation Memoization

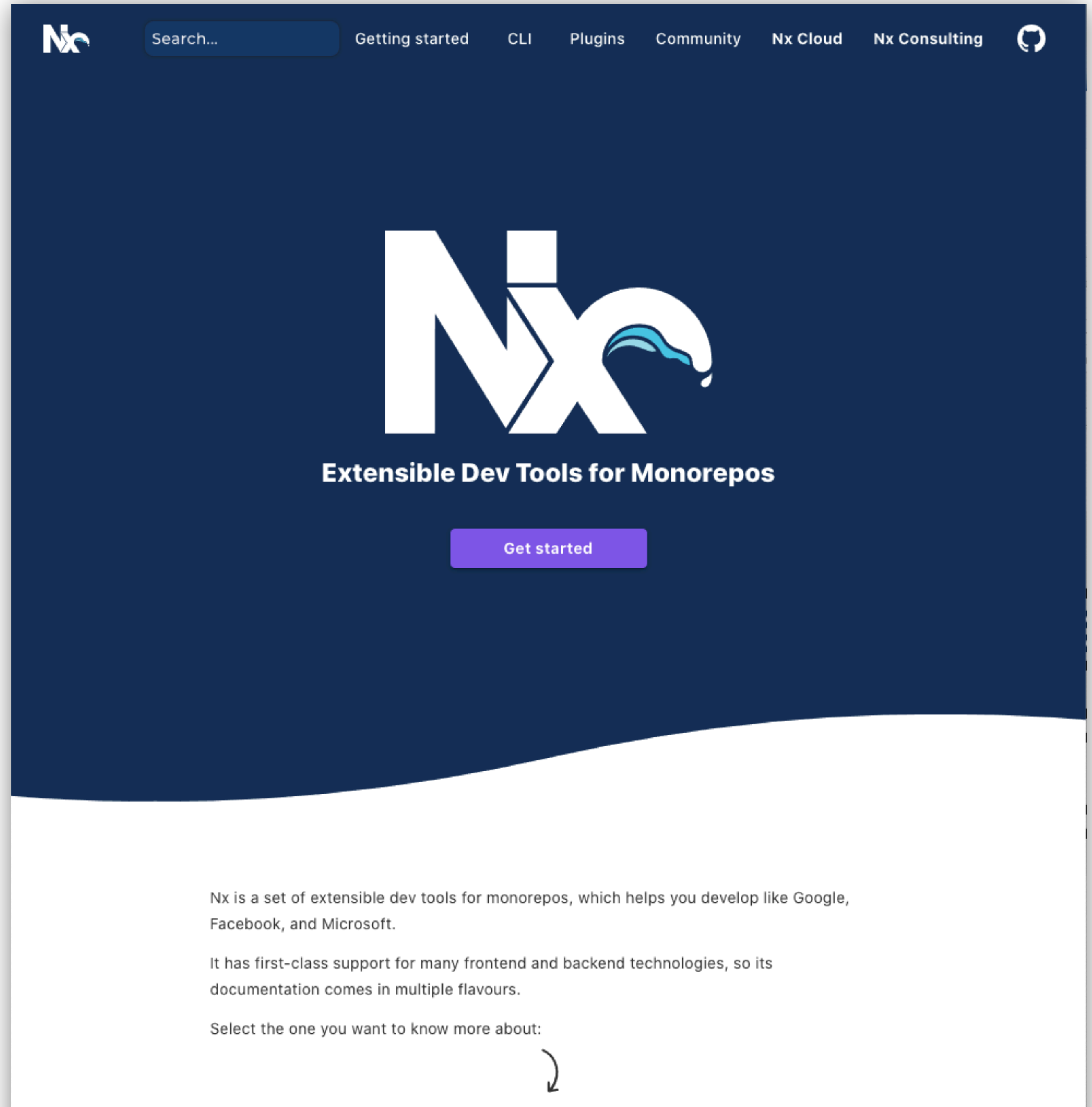


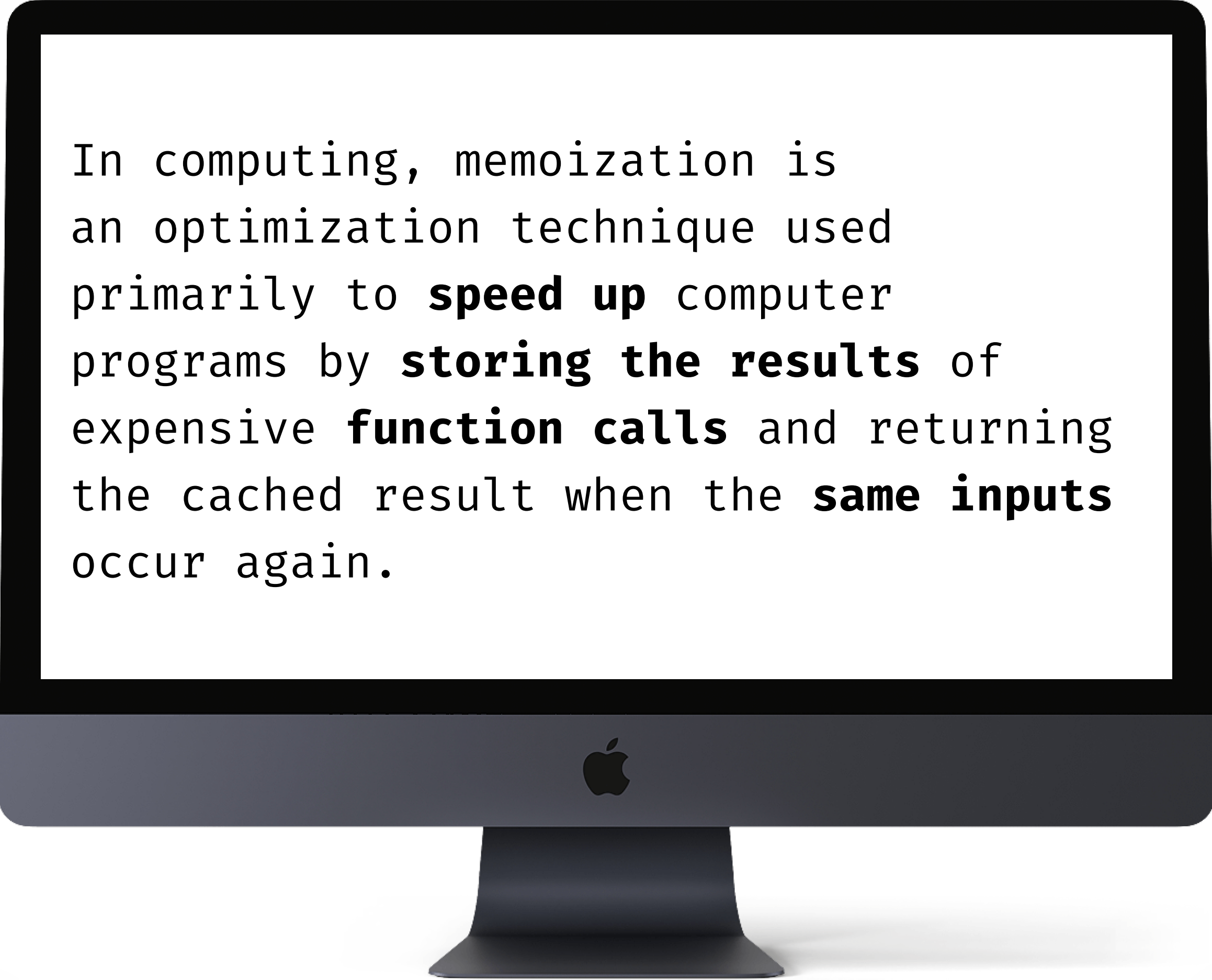
Make Angular CLI Faster

Ng Conf Colombia

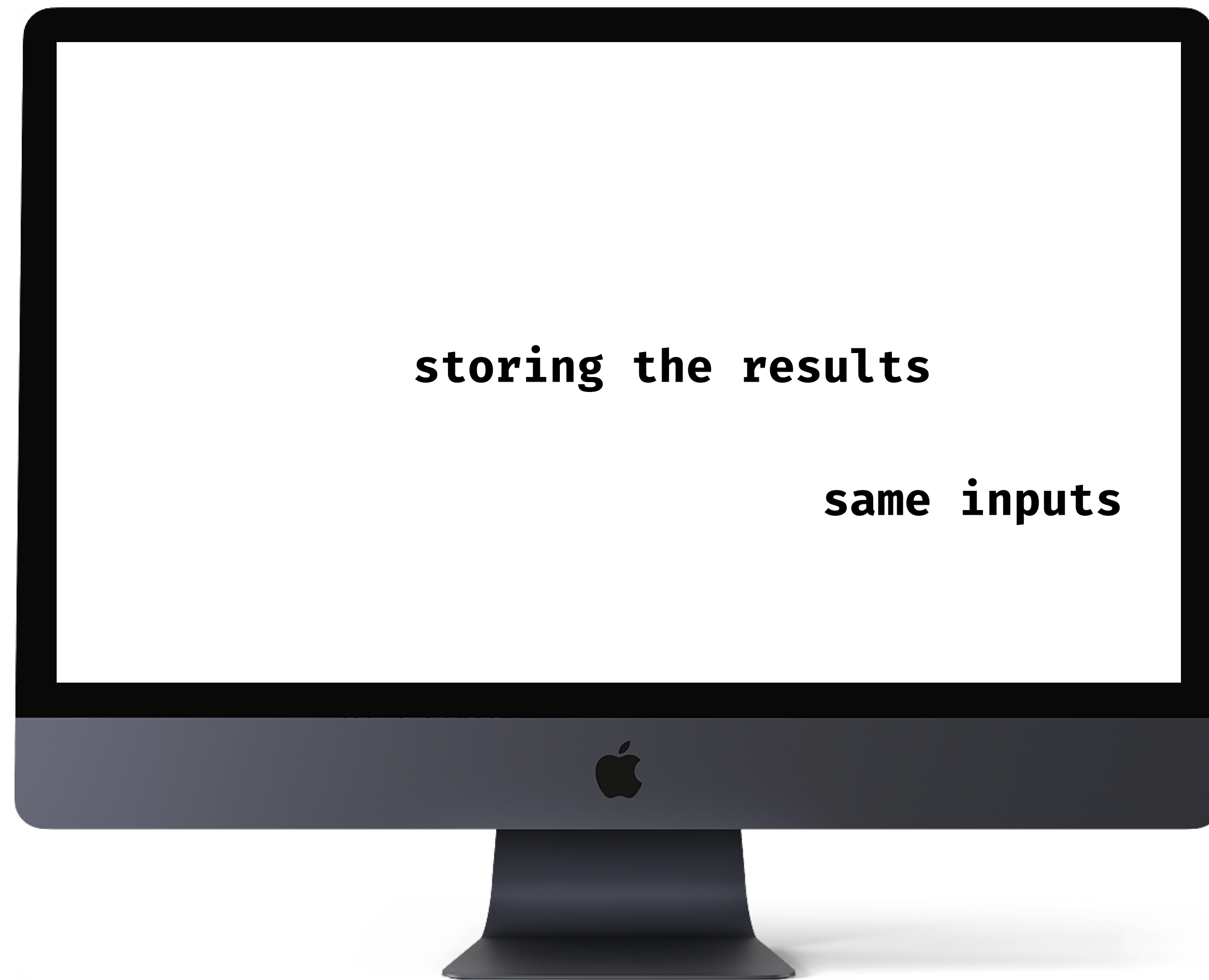
15 de agosto de 2020

nx.dev



A dark grey computer monitor with a black bezel and a black stand. The screen is white and displays text in a monospaced font. The text explains the concept of memoization in computing, highlighting key terms like 'speed up', 'storing the results', 'function calls', and 'same inputs' in bold.

In computing, memoization is an optimization technique used primarily to **speed up** computer programs by **storing the results** of expensive **function calls** and returning the cached result when the **same inputs** occur again.



Inputs

- **Commands** & args (i.e. `ng test`)
- Hashes!
 - Source Code
 - Runtime (node version, etc)
 - Implicit dependencies
 - Node modules

Results

- Output files
- Process output

Demo 1

Test a project, no cache hits

> **NX** **NOTE** Affected criteria defaulted to `--base=master --head=HEAD`

```
PASS packages/workspace/src/core/affected-project-graph/locators/implicit-json-changes.spec.ts
PASS packages/workspace/src/core/affected-project-graph/locators/npm-packages.spec.ts
PASS packages/workspace/src/command-line/print-affected.spec.ts
PASS packages/workspace/src/core/file-utils.spec.ts
PASS packages/workspace/src/core/affected-project-graph/locators/workspace-projects.spec.ts
PASS packages/workspace/src/core/project-graph/operators.spec.ts
PASS packages/workspace/src/utils/json-diff.spec.ts
PASS packages/workspace/src/tasks-runner/utils.spec.ts
PASS packages/workspace/src/utils/fileutils.spec.ts
PASS packages/workspace/src/core/project-graph/project-graph-builder.spec.ts
PASS packages/workspace/src/core/file-graph/file-map.spec.ts
PASS packages/workspace/src/core/normalize-nx-json.spec.ts
```

```
Test Suites: 74 passed, 74 total
Tests:       297 passed, 297 total
Snapshots:   0 total
Time:        24.548 s, estimated 33 s
Ran all test suites.
```

> **NX** **WARNING** Nx Cloud Problems

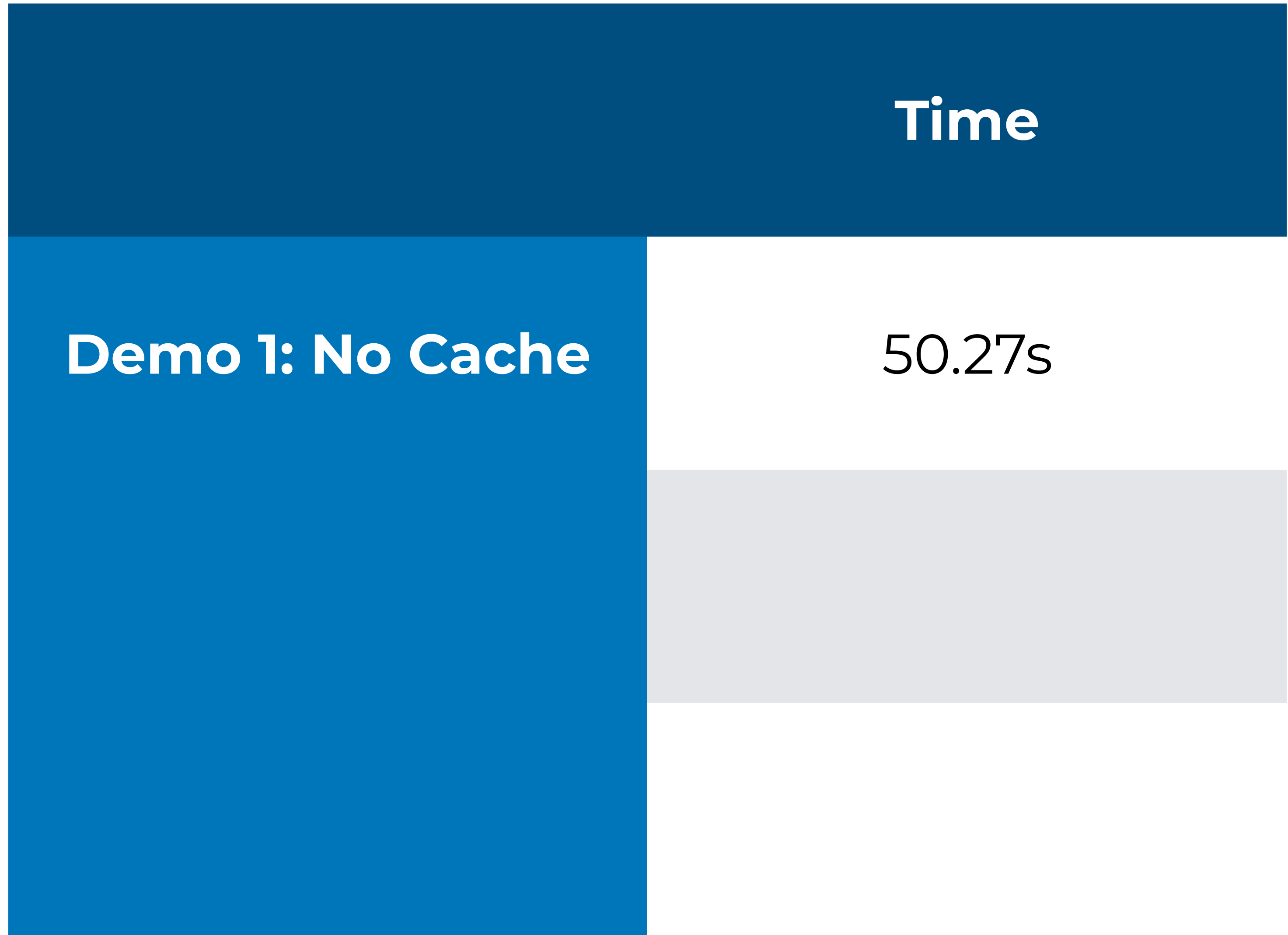
– Cannot connect to remote cache (scope: api, code: ENOTFOUND).

```
nx run workspace:test 50.27s user 9.52s system 230% cpu 25.938 total
```

```
$ █
```

Scores

Time to test



Demo 2

Test a project, local cache hits

```
PASS packages/workspace/src/core/affected-project-graph/locators/implicit-json-changes.spec.ts
PASS packages/workspace/src/schematics/move/lib/check-destination.spec.ts
PASS packages/workspace/src/schematics/move/lib/update-project-root-files.spec.ts
PASS packages/workspace/src/schematics/remove/lib/update-nx-json.spec.ts
PASS packages/workspace/src/migrations/update-8-2-0/update-8-2-0.spec.ts
PASS packages/workspace/src/core/file-graph/file-map.spec.ts
PASS packages/workspace/src/schematics/remove/lib/update-tsconfig.spec.ts
PASS packages/workspace/src/utils/fileutils.spec.ts
PASS packages/workspace/src/schematics/move/lib/move-project.spec.ts
PASS packages/workspace/src/schematics/remove/lib/remove-project.spec.ts
PASS packages/workspace/src/schematics/move/lib/update-nx-json.spec.ts
PASS packages/workspace/src/command-line/print-affected.spec.ts
PASS packages/workspace/src/schematics/run-commands/run-commands.spec.ts
PASS packages/workspace/src/utils/rules/check-project-exists.spec.ts
PASS packages/workspace/src/utils/buildable-libs-utils.spec.ts
PASS packages/workspace/src/core/normalize-nx-json.spec.ts
PASS packages/workspace/src/utils/cli-config-utils.spec.ts
PASS packages/workspace/src/utils/common.spec.ts
PASS packages/workspace/src/migrations/update-10-1-0/migrate-eslintrc-tsconfig-wildcard.spec.ts
PASS packages/workspace/src/schematics/workspace-schematic/workspace-schematic.spec.ts
```

Test Suites: 74 passed, 74 total

Tests: 297 passed, 297 total

Snapshots: 0 total

Time: 35.135 s

Ran all test suites.

nx run workspace:test 0.49s user 0.12s system 67% cpu 0.899 total

\$ █

Scores

Time to test

	Time
Demo 1: No Cache	50.27s
Demo 2: Local Cache	0.49s

Demo 3

Test a project, **remote** cache hits



```
PASS packages/workspace/src/core/affected-project-graph/locators/implicit-json-changes.spec.ts
PASS packages/workspace/src/schematics/move/lib/check-destination.spec.ts
PASS packages/workspace/src/schematics/move/lib/update-project-root-files.spec.ts
PASS packages/workspace/src/schematics/remove/lib/update-nx-json.spec.ts
PASS packages/workspace/src/migrations/update-8-2-0/update-8-2-0.spec.ts
PASS packages/workspace/src/core/file-graph/file-map.spec.ts
PASS packages/workspace/src/schematics/remove/lib/update-tsconfig.spec.ts
PASS packages/workspace/src/utils/fileutils.spec.ts
PASS packages/workspace/src/schematics/move/lib/move-project.spec.ts
PASS packages/workspace/src/schematics/remove/lib/remove-project.spec.ts
PASS packages/workspace/src/schematics/move/lib/update-nx-json.spec.ts
PASS packages/workspace/src/command-line/print-affected.spec.ts
PASS packages/workspace/src/schematics/run-commands/run-commands.spec.ts
PASS packages/workspace/src/utils/buildable-libs-utils.spec.ts
PASS packages/workspace/src/core/normalize-nx-json.spec.ts
PASS packages/workspace/src/utils/rules/check-project-exists.spec.ts
PASS packages/workspace/src/utils/cli-config-utils.spec.ts
PASS packages/workspace/src/utils/common.spec.ts
PASS packages/workspace/src/migrations/update-10-1-0/migrate-eslintrc-tsconfig-wildcard.spec.ts
PASS packages/workspace/src/schematics/workspace-schematic/workspace-schematic.spec.ts
```

Test Suites: **74 passed**, 74 total

Tests: **297 passed**, 297 total

Snapshots: 0 total

Time: 48.62 s

Ran all test suites.

nx run workspace:test 0.49s user 0.12s system 38% cpu 1.605 total

\$ █

Scores

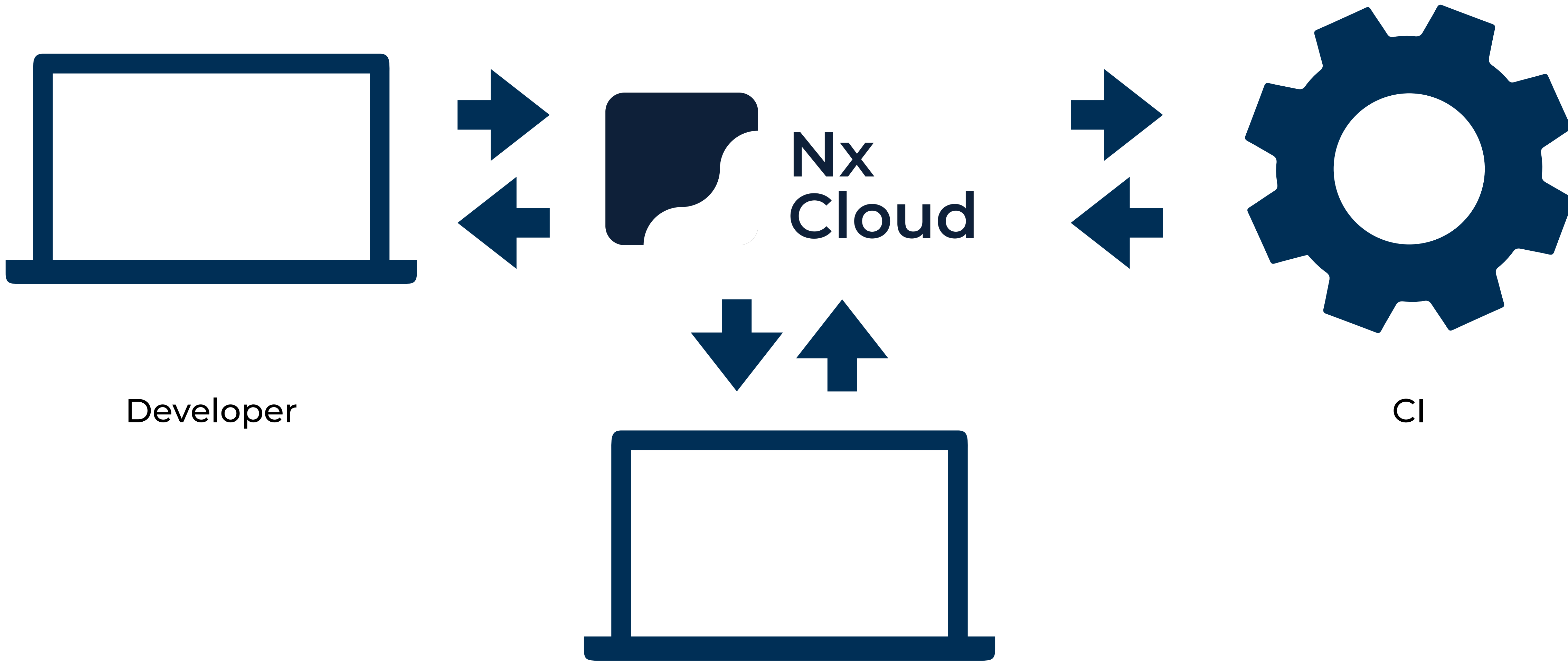
Time to test

	Time
Demo 1: No Cache	50.27s
Demo 2: Local Cache	0.49s
Demo 3: Teammate	0.49s

HOW?!?!?



nx.app

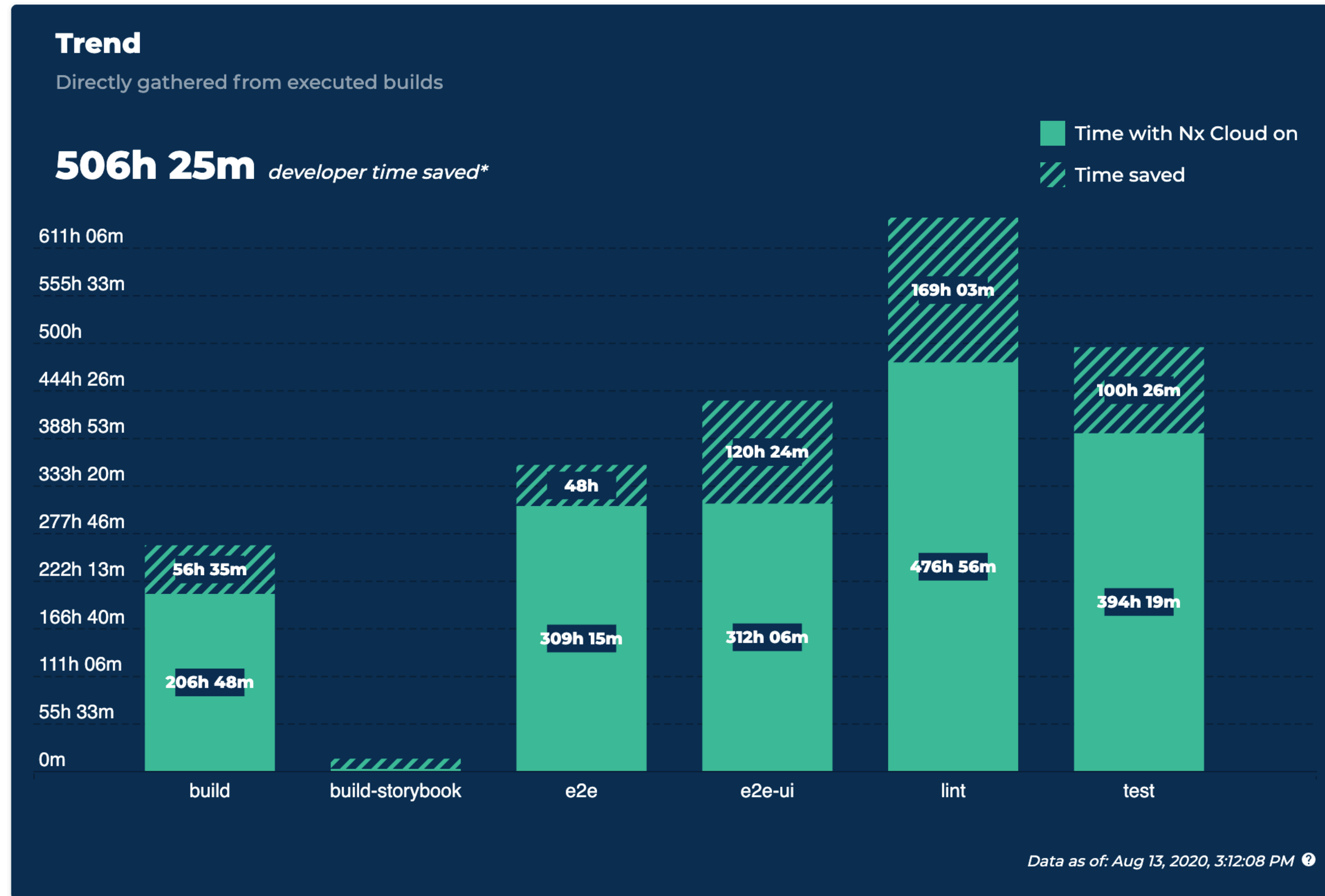


Developer

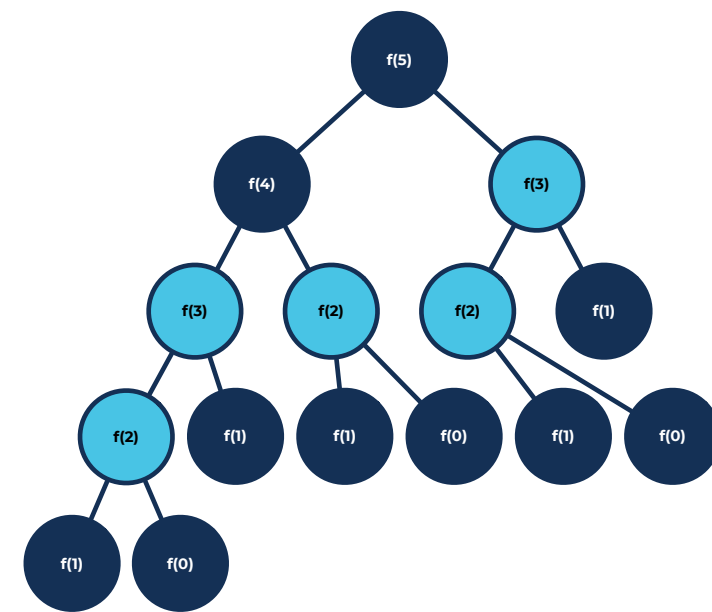
CI

Developer





Data from a real workspace with all contributors connected to Nx Cloud's distributed computation cache



Memoization

What is it?



Nx's Computation Memoization

How Nx memoizes computations



Make Angular CLI Faster

How to get faster builds with Angular CLI by using computation memoization



Make Angular CLI Faster

How to get faster
builds with Angular
CLI by using
computation
memoization

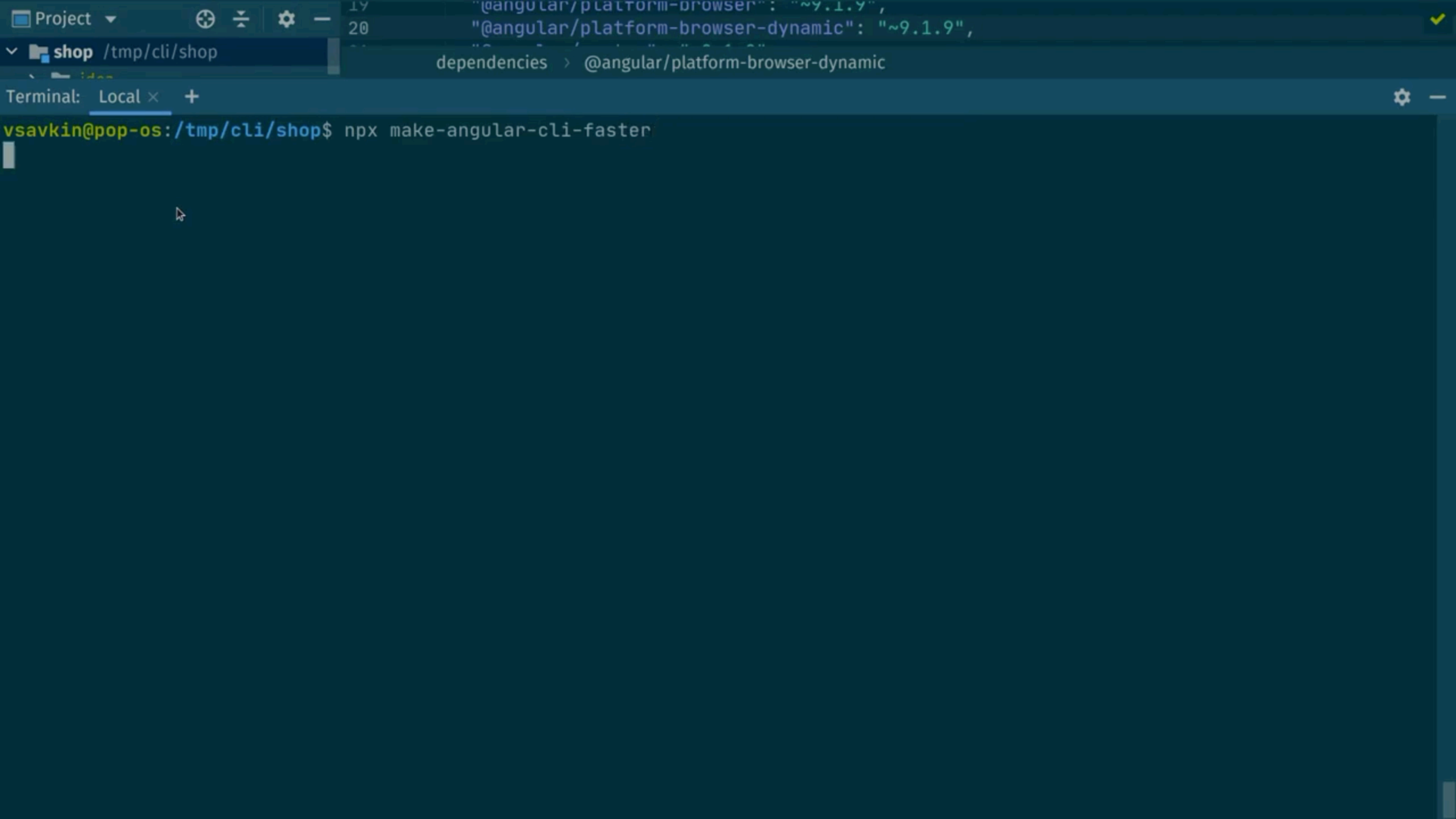
Make Angular CLI Faster

Ng Conf Colombia

15 de agosto de 2020

Make Angular CLI Faster

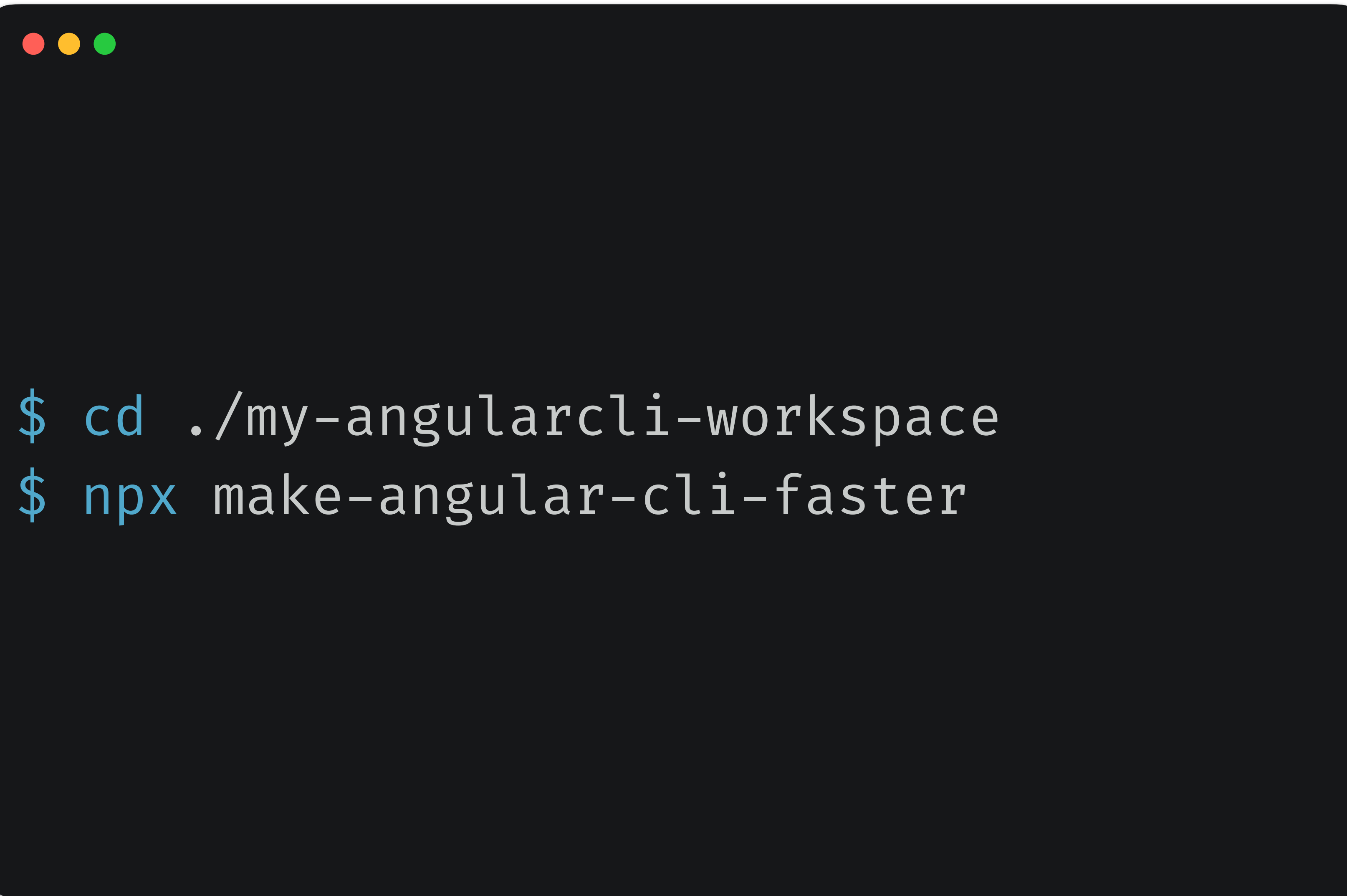




Non-Invasive

Bare minimum of Nx to make Angular CLI faster

- Add an nx.json configuration
- Creates a free Nx Cloud workspace
- Decorates Angular CLI to utilize cache
- **Once changes are checked in, builds get up to 10x faster for your whole team and CI**



```
$ cd ./my-angularcli-workspace
$ npx make-angular-cli-faster
```