CS-121

Introduction to Programming

 With Python

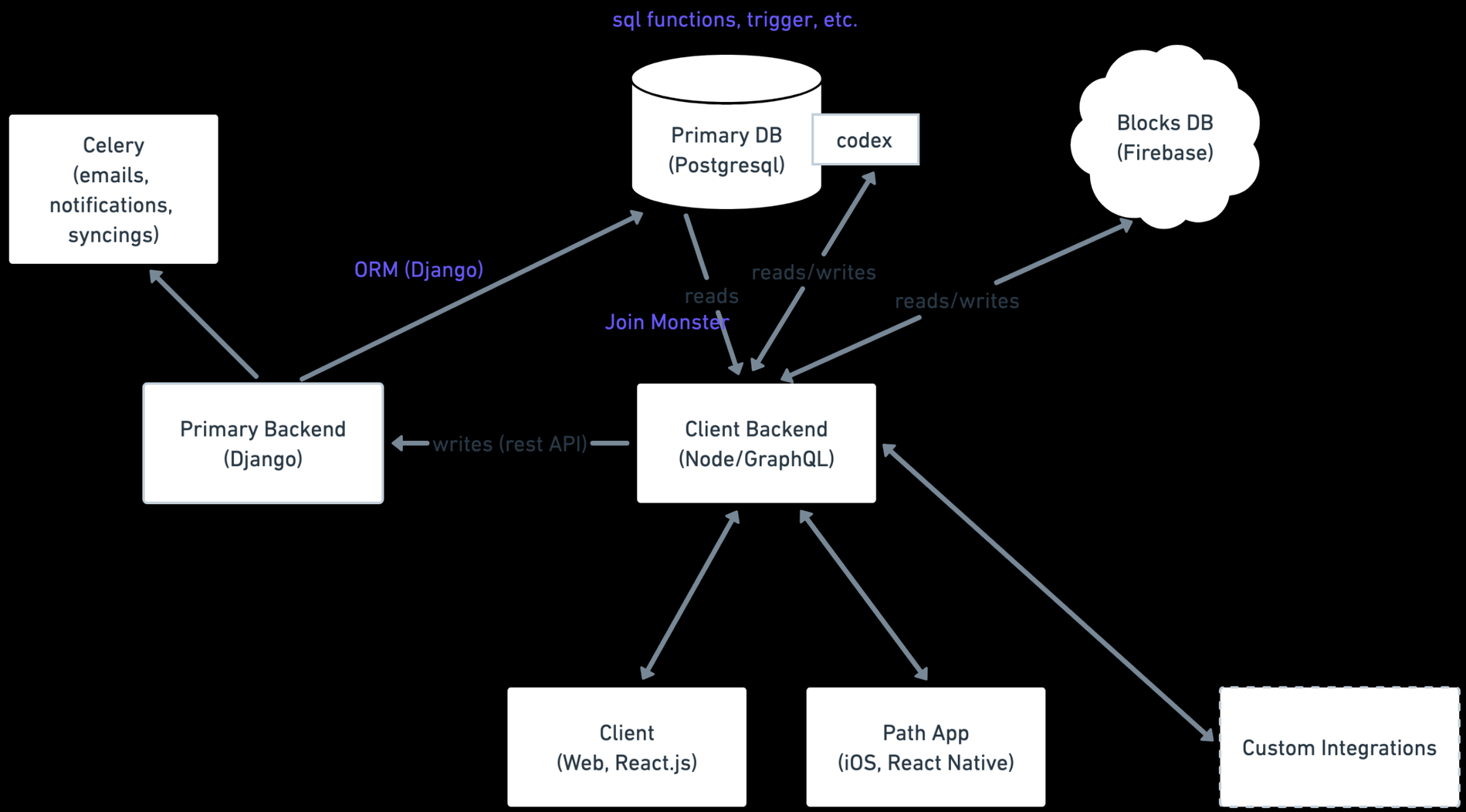
→ CTO/Founder Pathwright

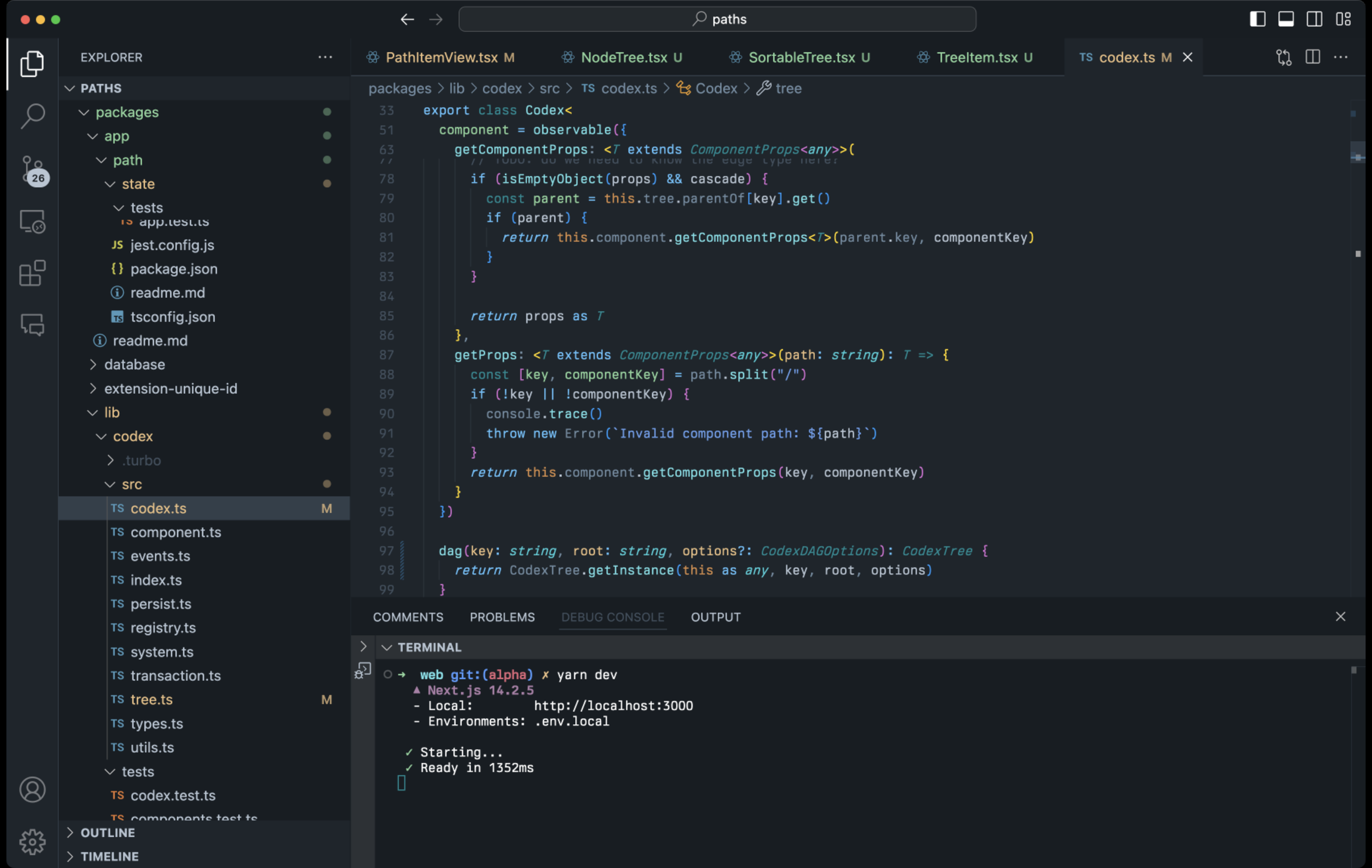
→ Design Engineer

→ Teacher


BUSINESS
BLACK BOX
2012
WE ARE PATHWRIGHT
THE IRON YARD
BREEDING GROUNDS FOR THE NEXT BIG THING
CHANGE THE GAME
MAKE A NEW PATH
R.I.G.HT HERE
IN UPSTATE SC
LEARNING TODAY IS THE SAME.
EDUCATING IS DIFFERENT.
READ MORE INSIDE!
U.S. $5.95


LEARN
DEATH
PATHWRIGHT
UNTIL
DEATH
PATHWRIGHT
UNTIL
DEATH
PATHWRIGHT


sql functions, trigger, etc.
Primary DB
(Postgresql)
codex
Blocks DB
(Firebase)
Celery
(emails,
notifications,
syncings)
ORM (Django)
reads
reads/writes
reads/writes
Join Monster
Primary Backend
(Django)
writes (rest API)
Client Backend
(Node/GraphQL)
Client
(Web, React.js)
Path App
(iOS, React Native)
Custom Integrations

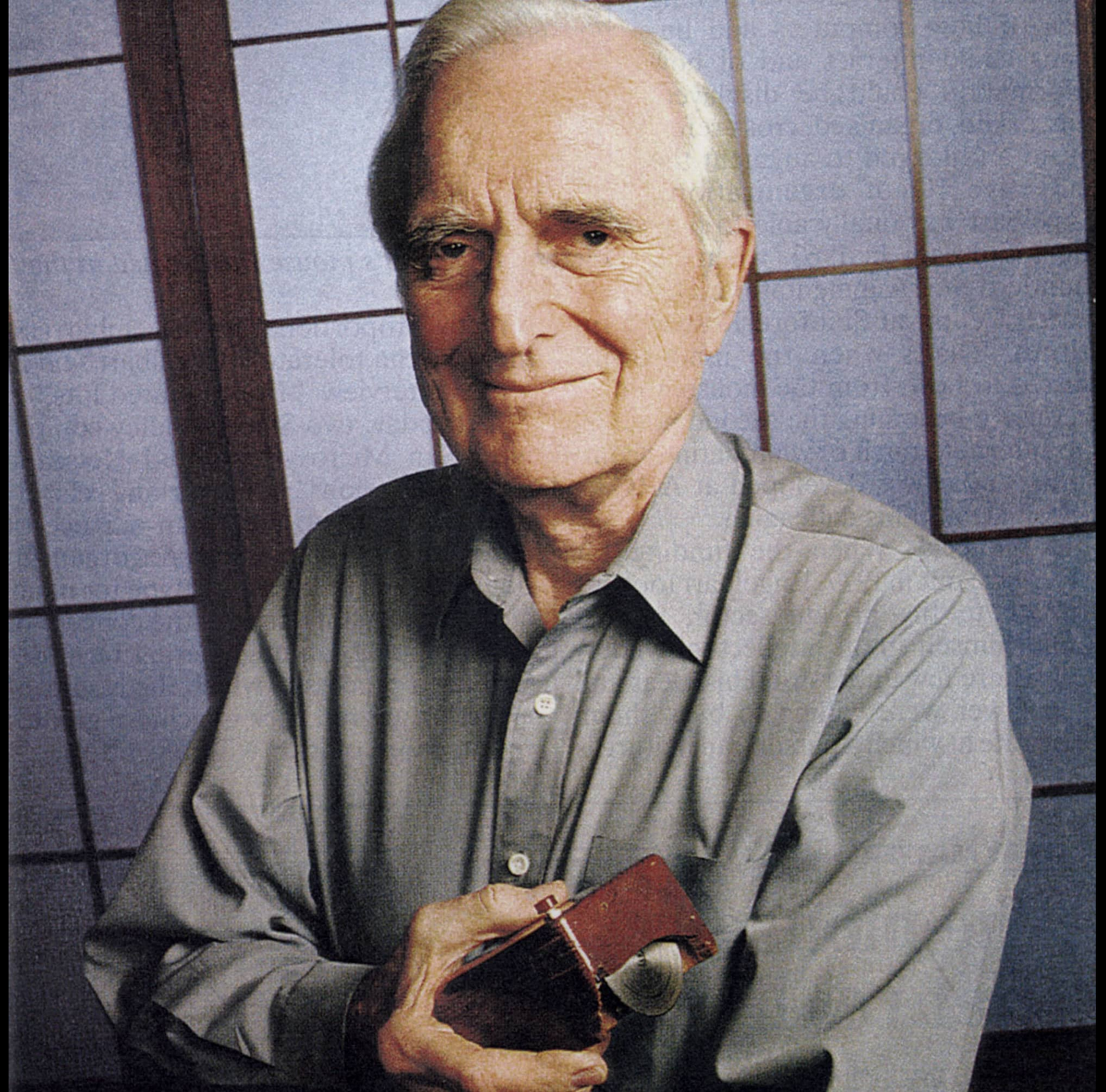

paths
EXPLORER
PATHS
packages
app
path
state
tests
app.test.ts
jest.config.js
package.json
readme.md
tsconfig.json
readme.md
database
extension-unique-id
lib
codex
.turbo
src
codex.ts M
component.ts
events.ts
index.ts
persist.ts
registry.ts
system.ts
transaction.ts
tree.ts M
types.ts
utils.ts
tests
codex.test.ts
components.test.ts
OUTLINE
TIMELINE
PathItemView.tsx M
NodeTree.tsx U
SortableTree.tsx U
TreeItem.tsx U
codex.ts M
packages > lib > codex > src > codex.ts > Codex > tree
export class Codex<
component = observable({
getComponentProps: <T extends ComponentProps<any>>(
// TODO: do we need to know the edge type here?
if (isEmptyObject(props) && cascade) {
const parent = this.tree.parentOf[key].get()
if (parent) {
return this.component.getComponentProps<T>(parent.key, componentKey)
}
}
return props as T
},
getProps: <T extends ComponentProps<any>>(path: string): T => {
const [key, componentKey] = path.split("/")
if (!key || !componentKey) {
console.trace()
throw new Error(`Invalid component path: ${path}`)
}
return this.component.getComponentProps(key, componentKey)
}
})
dag(key: string, root: string, options?: CodexDAGOptions): CodexTree {
return CodexTree.getInstance(this as any, key, root, options)
}
COMMENTS
PROBLEMS
DEBUG CONSOLE
OUTPUT
TERMINAL
web git:(alpha) ✗ yarn dev
▲ Next.js 14.2.5
- Local: http://localhost:3000
- Environments: .env.local
✓ Starting...
✓ Ready in 1352ms

What is computer programming?

Computer programming, also known as coding, is the process of writing instructions for computers to follow, which are then compiled into programs.

What can you *do* with computer programming?

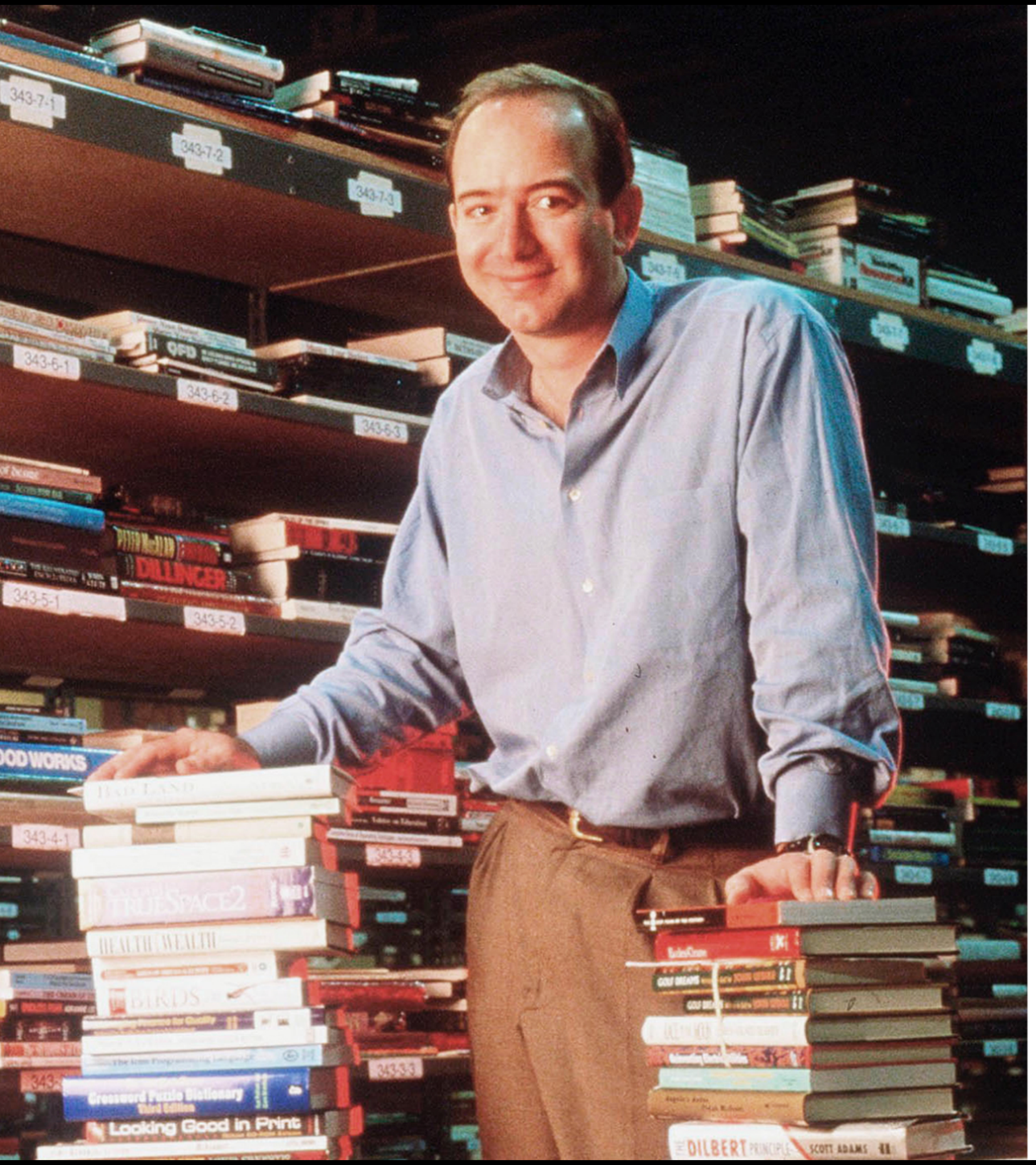
Computer Programers




MICROSOFT
minidisk
Verbatim
This Side Up


CERN
World Wide Web Initiative
EIZO


DO YOU HAVE
CHANGE FOR
PARADIGM?


343-7-1
343-7-2
343-7-3
343-6-1
343-6-2
343-6-3
343-5-1
343-5-2
343-4-1
343-3-3
DILLINGER
TRUESPACE2
BIRDS
Looking Good in Print
THE DILBERT PRINCIPLE
SCOTT ADAMS


Jeff
Bezos


Google!

Co-Founders of YouTube Sell To Google for $1.65 Billion in 2006

You?

What did they all *do* with code?

They all **solved problems** for humans

Why Code?

To **solve problems for humans**

“Coding is the closest thing we have to a superpower”

“You might not think that programmers are artists, but programming is an extremely creative profession. It's logic-based creativity.”

“Every great developer you know got there by solving problems they were unqualified to solve until they actually did it.”

How to learn to code?

By coding!

No one* can learn to code for you

Where can school help?

→ Theory

→ Practice

→ Projects

Where might school fall short?

→ To much theory

→ Not enough practice

→ Unrealistic projects

What to expect from this class

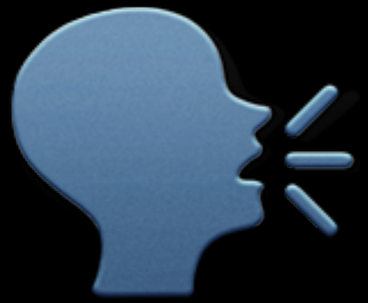
80% code

20% theory

100% projects

What does learning *feel* like?

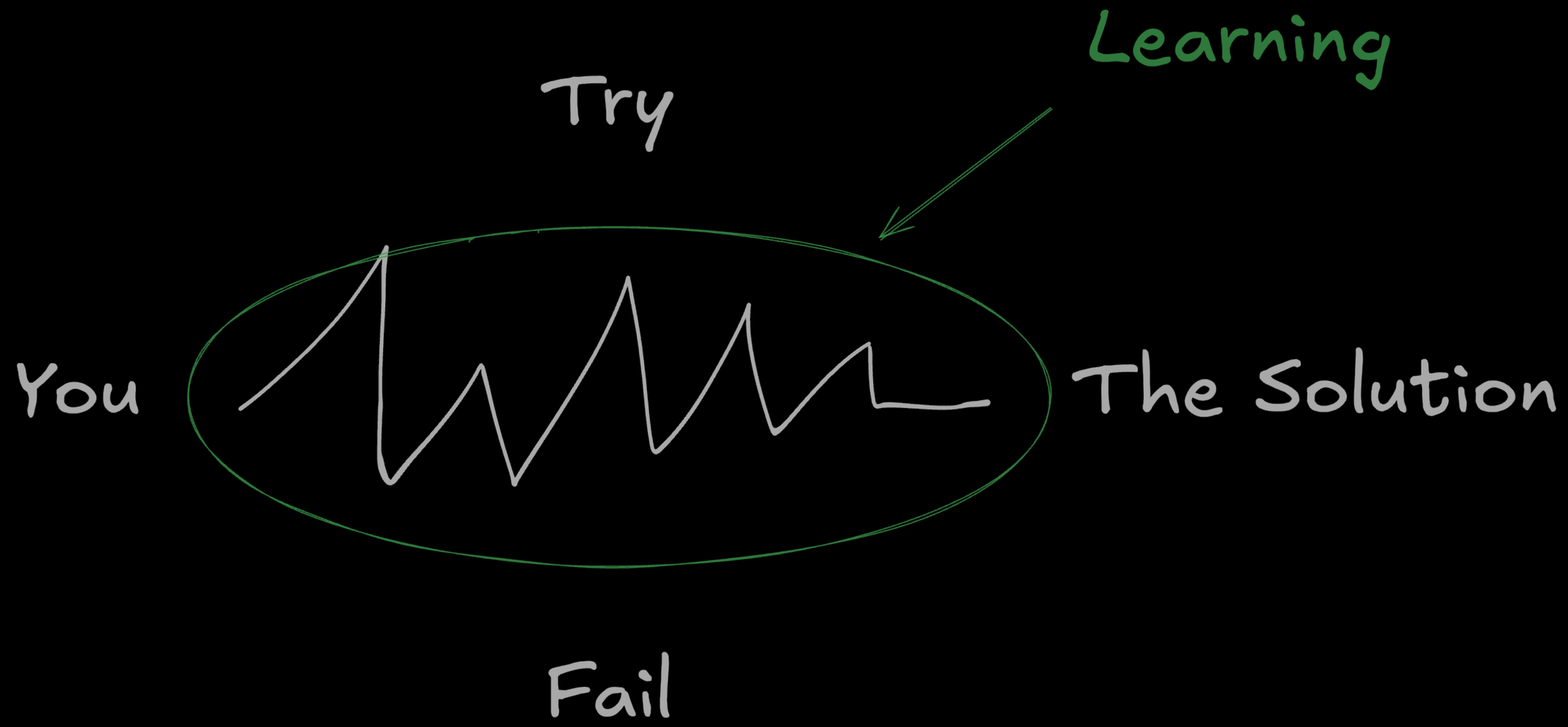

Unkown
Area
Foggy
Area
Fully Visible
Area



Try

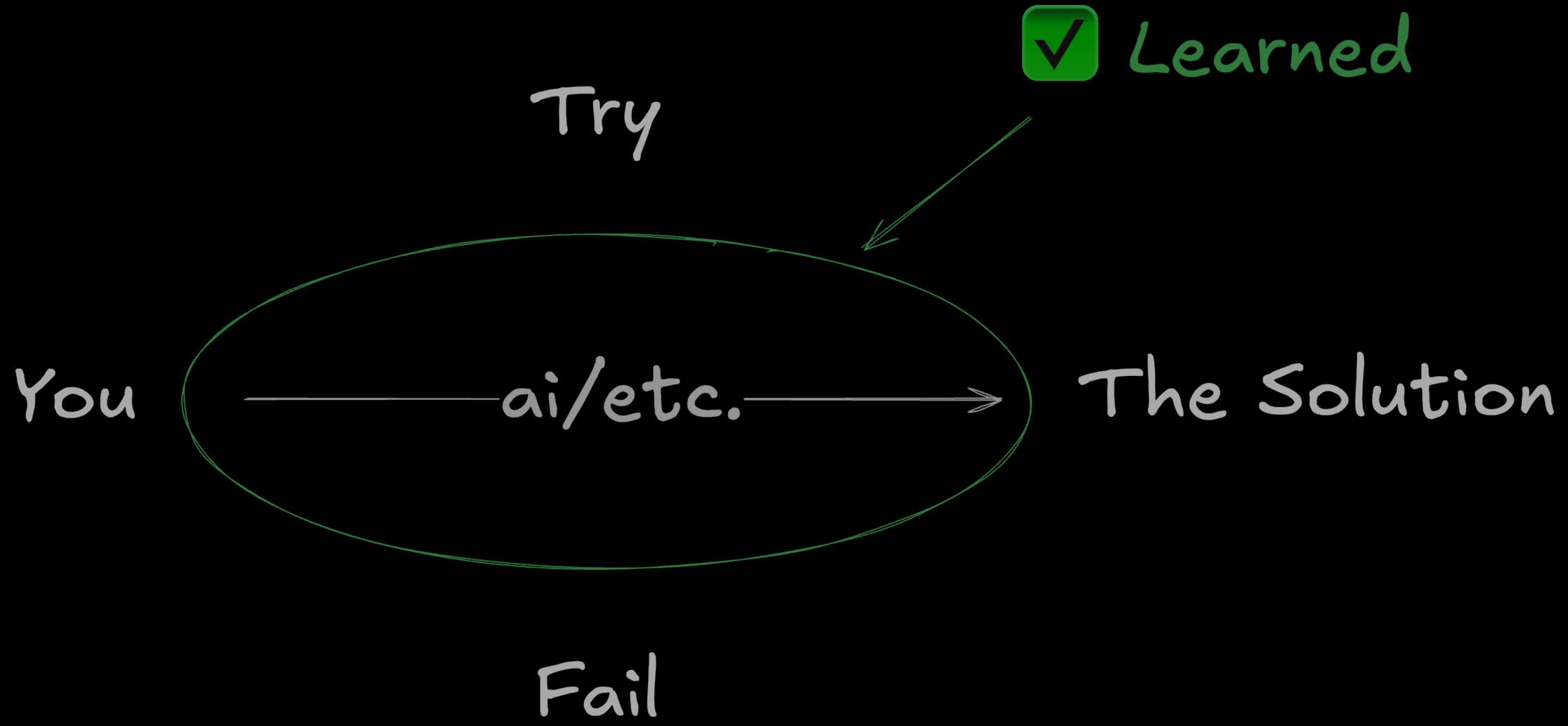
You  The Solution

Fail


Learning
Try
You
The Solution
Fail

Learning?

You ——AI/Copy/Paste——→ The Solution


Learned
Try
You
ai/etc.
The Solution
Fail

Immersion + practice

Where does theory fit in?

→ Understanding the "why"

→ "Learning your scales"

→ Not necessarily going to help you make anything

Grading

Projects (50%)

→ Solve real problems

→ Build real things

→ Choose your own adventure (within reason)

Practice (25%)

→ Lab projects

→ Test projects

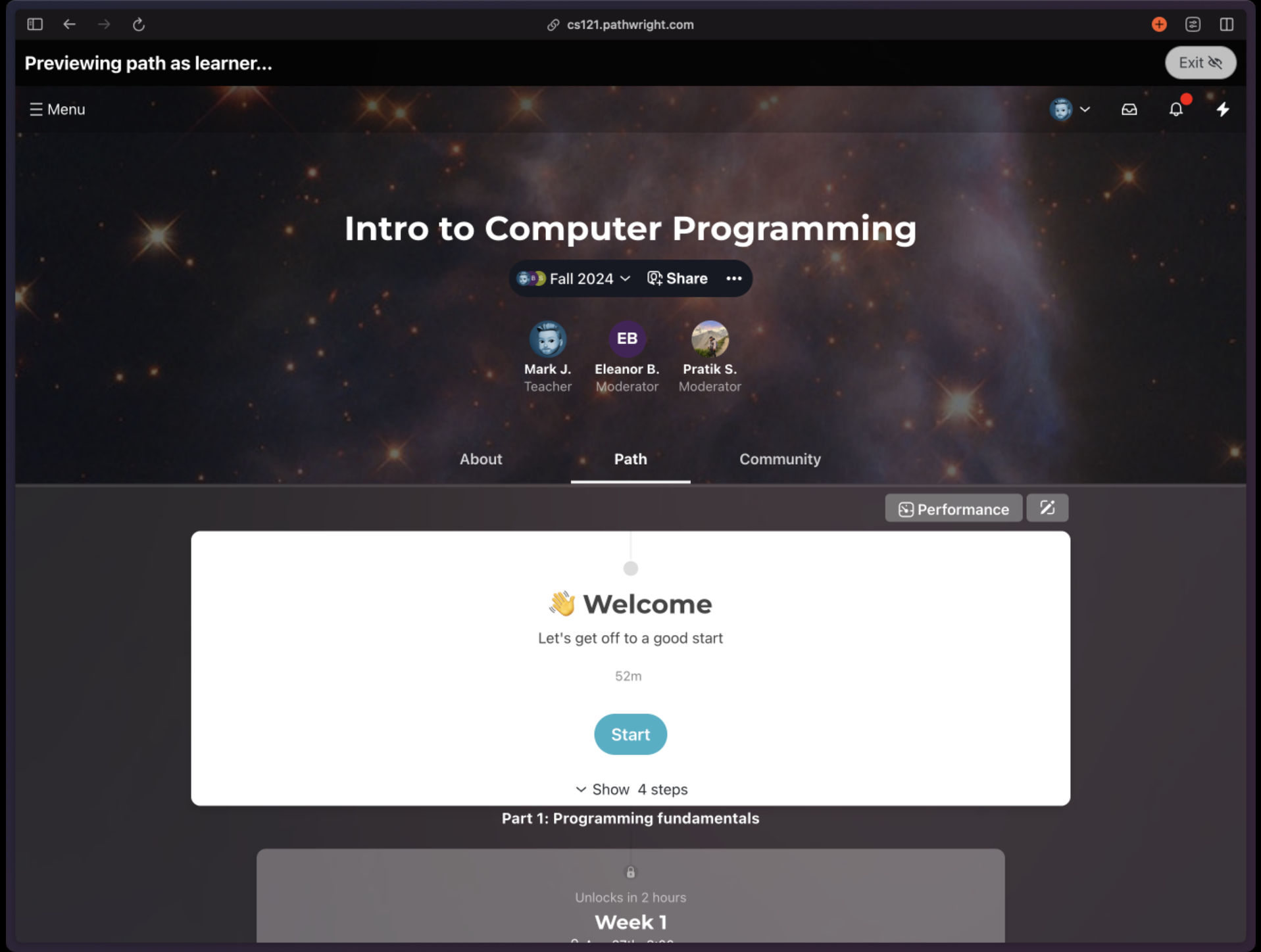
→ Code Challenges

Professionalism (25%)

→ "Showing up"

→ On time (work and attendence)

→ Clear Communication and documentation


cs121.pathwright.com
Previewing path as learner...
Exit
Menu
Intro to Computer Programming
Fall 2024
Share
Mark J.
Teacher
EB
Eleanor B.
Moderator
Pratik S.
Moderator
About
Path
Community
Performance
Welcome
Let's get off to a good start
52m
Start
Show 4 steps
Part 1: Programming fundamentals
Unlocks in 2 hours
Week 1

🔬 Lab

→ On Zoom: 2:30 pm on Wed.

→ Lab projects

→ Completion-based

🧭 The Course Path

→ Step-by-step

→ Updated every week

→ Weekly Lab + Project

→ Due dates and notifications

→ Learning resources

💡 Path Tips

→ Mark every step as complete (including lectures)

→ Use the discussion below any step to ask questions

→ Add your own notes to the path if you like

Getting help

AI?

Challenges

→ Afternoon slump

→ Unfamiliar tools and environments

→ Frustration

→ Real learning is uncomfortable

→ We are here to help

→ It does get easier

Accept the initial discomfort.

Embrace the fog of war.

Keep going. Never give up.


bitly