



Aleppo js

JS MEETUP - 6 SEPTEMBER 2025



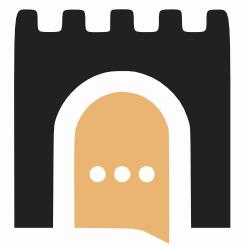
**OPENING WORD**



# SPONSOR



[elwaseet-sy.com](http://elwaseet-sy.com)



SPONSOR

الهيئة  
الطلابية  
محافظة حلب



# MODERN FRONTEND PROJECT STRUCTURE



# ABOUT ME



**Mohammed Taher Ghazal**  
**A Senior Front-end Developer**



[linkedin.com/in/mtg-software-dev](https://linkedin.com/in/mtg-software-dev)

# WHAT WILL WE COVEER TODAY

- Choosing the right tech stack
- Separation of concerns
- Benefiting from automation tools to maintain consistency
- Should I use AI? When and where?
- Monorepos
- Q&A

# WHY IS THIS TOPIC IMPORTANT

- Crucial upcoming stage for Syria
- New projects and infrastructure needs
- New remote working opportunities
- A majority of our users are not tech-proficient

## FEW THINGS TO KEEP IN MIND

- No one size fits all (Different project scopes, team size, team skill level, personal preferences)
- It's an iterative process
- Constantly evolving/learning

# CHOOSING THE RIGHT STACK

- Bigger area of impact → more consideration needed
- How stable is it (LTS, Breaking changes frequency, ...etc)
- How frequently does it get updated (solved issues, open PRs, ...etc)
- How many developers around can use it (Hiring, community support, onboarding)

⚠ It's important that you maintain your dependencies up-to-date.

# SEPARATING PROJECT LAYERS

- There are several methods for separating a project into layers (DDD, Clean architecture, IoC, ...etc)
- Keep your business logic separated from your app/UI logic
- Don't over-abstract
- Make it work → Make it good

# **SEPARATING PROJECT LAYERS**

## **Main Benefits:**

- Reusability
- Scalability
- Easier to test and debug
- Easier to validate by non-frontend devs or even non-technical people

# **RULES OF THE HOUSE**

## **The Problem**

- Differences in personal preferences
- Differences in skills & exp levels
- Not feasible to manually review every developers work for smaller details
- Creates extra diff lines due to inconsistent formatting

# RULES OF THE HOUSE

## When automated

- Eliminates a big category of errors by catching them before reaching production
- Increases codebase quality and cleanliness
- Quicker reviews & merging iteration
- Some of the tools include: ESLint, Biome, TypeScript, Prettier, Husky, Github Actions

# AI... TO USE OR NOT TO USE

- You have to learn the fundamentals and do them
- Not for all types of tasks
- Have to review/learn/update (a lot)
- Treat as a junior developer, otherwise just as some pointers

**MONOREPOS**

# QUICK RECAP

- Spend time choosing the tech stack
- Separation of concerns
- Leverage automation tools to maintain consistency
- Use AI responsibly (otherwise minimize)
- Learn about monorepos

# TO READ MORE



[linkedin.com/in/mtg-software-dev](https://linkedin.com/in/mtg-software-dev)



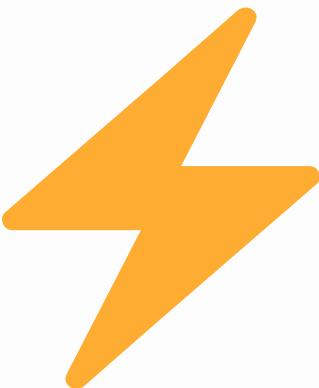
[mtg-dev.tech/blog](https://mtg-dev.tech/blog)

Q & A time



adc

NETWORKING  
BREAK



# LIGHTNING TALKS

# KNIP: DECLUTTER YOUR JAVASCRIPT PROJECTS

Clean Code for Better Performance



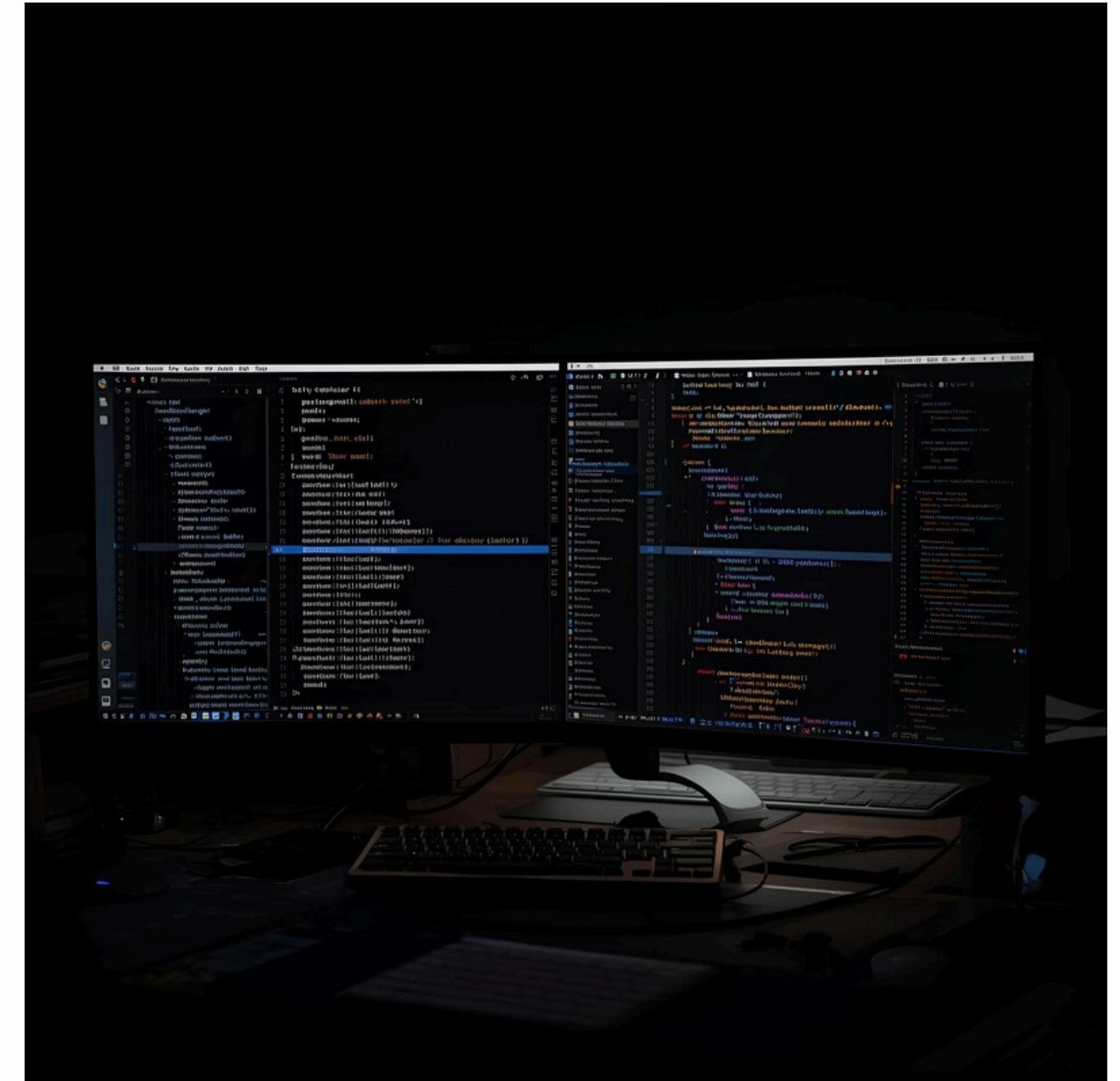
Nour Samman

Front-end developer



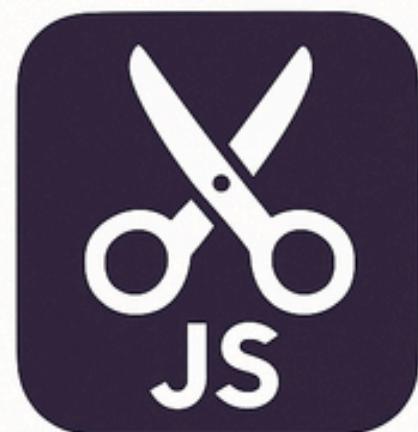
# THE CHALLENGES WE FACE

- Larger bundles
- Slower build times
- Forgotten dependencies in package.json
- Unused files & functions



# WHAT IS KNIP

- Open-source tool for JavaScript and TypeScript projects
- Analyze the entire codebase to detect dead code and unnecessary elements
- detects :
  - Unused files
  - Unused exports
  - Unused dependencies



**knip**



# THE CODE BEFORE KNIP

- **Unused files (12)**
  - **Unused dependencies (8)**
  - **Unused devDependencies (4)**
  - **Unlisted dependencies (1)**
  - **Unused exports (16)**
  - **Unused exported types (2)**

# YOUR WEBSITE, BETTER

```
PS C:\Users\Mohammed Nour Samman\Des  
>>  
✖ Excellent, Knip found no issues.
```

Use Knip to achieve optimal performance.

Connect with me on LinkedIn:  
[in/nour-samman](https://www.linkedin.com/in/nour-samman)

# DOCUMENTING YOUR JAVASCRIPT

# WHY DOCUMENTATION?

- Faster code understanding
- Reduced errors
- Saved time
- Project continuity

looking for  
documentation: making  
documentation:



# TYPES OF DOCUMENTATION

- 1. External (README.md, Wiki )
- 2. Internal (Comments, Type Hints)
- 3. Generated (JSDoc, Typedoc, Swagger )

# TOOLS

## JSDOC

- Documentation generator for JavaScript/TypeScript
- Add comments above functions/classes
- Generate HTML docs
- Improves IntelliSense in editor



Visit link to learn more :  
[jsdoc.app](https://jsdoc.app)

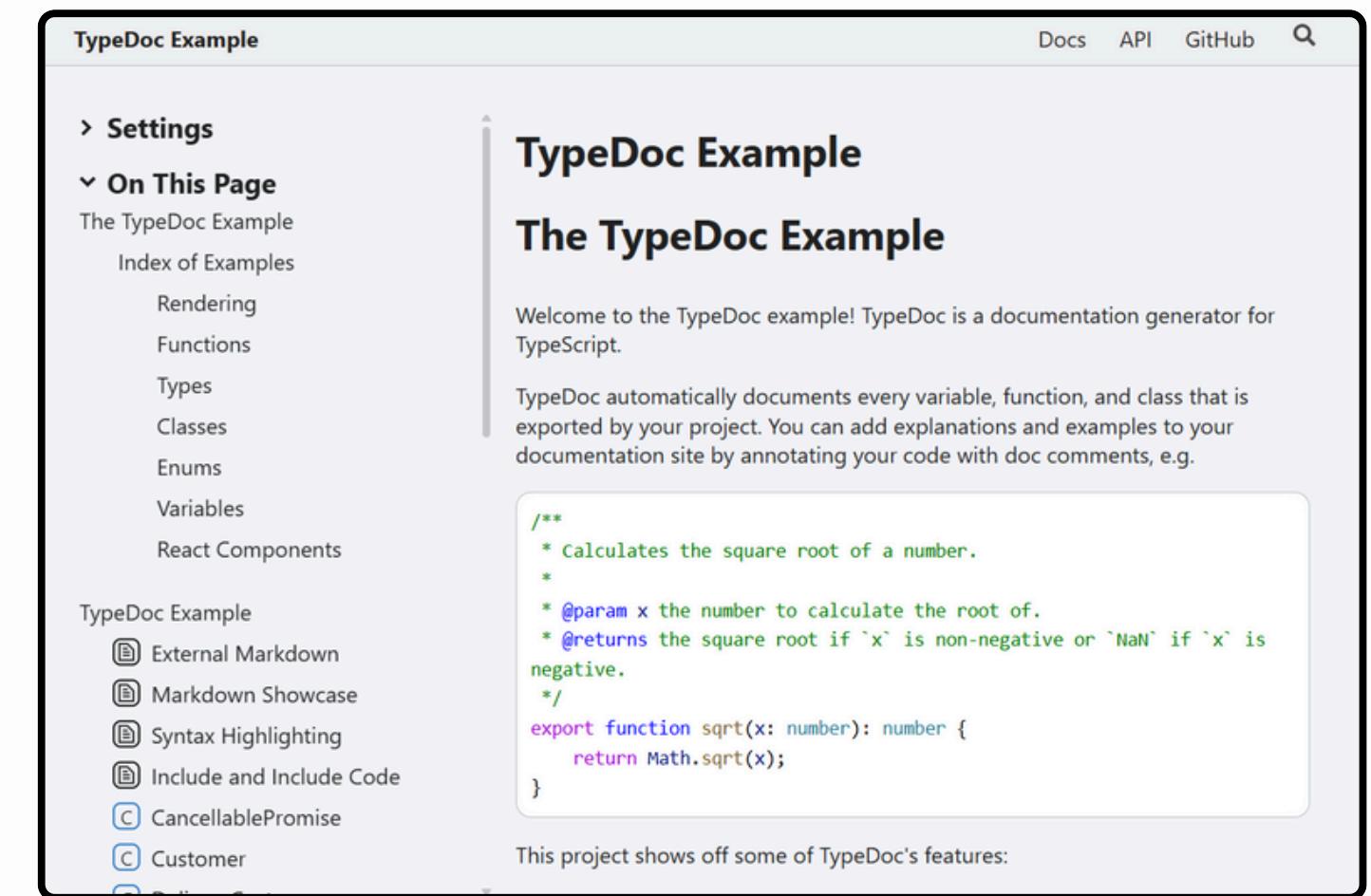
```
1  /**
2   * Adds two numbers together
3   * @example
4   * // returns 15
5   * methodExample(5, 10);
6   * @returns {Number} Returns the sum of a and b.
7   */
8  export const methodExample = (a: number, b: number) => {
9    return a + b;
10};
```

```
const methodExample: (a: number, b: number) => number
Adds two numbers together
@example
// returns 15
methodExample(5, 10);
@returns — Returns the sum of a and b.
methodExample(5, 10);
```

# TOOLS

## TYPERDOC

- **Reads TS types (interfaces, enums, generics)**
- **Generates clean HTML docs**
- **Great for TS-heavy projects**



The screenshot shows the TypeDoc Example website. The header includes links for Docs, API, GitHub, and a search icon. The left sidebar has a 'TypeDoc Example' heading and a 'On This Page' section with links to Settings, Rendering, Functions, Types, Classes, Enums, Variables, and React Components. Below this is a 'TypeDoc Example' section with a list of files: External Markdown, Markdown Showcase, Syntax Highlighting, Include and Include Code, CancellablePromise, Customer, and a file named 'File'. The main content area features a heading 'TypeDoc Example' and 'The TypeDoc Example'. It includes a welcome message, a description of how TypeDoc automatically documents variables, functions, and classes, and a code snippet showing a TypeScript function for calculating square roots.

TypeDoc Example

Docs API GitHub 

> Settings

On This Page

The TypeDoc Example

Index of Examples

Rendering

Functions

Types

Classes

Enums

Variables

React Components

TypeDoc Example

External Markdown

Markdown Showcase

Syntax Highlighting

Include and Include Code

CancellablePromise

Customer

File

TypeDoc Example

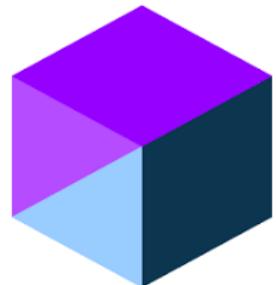
The TypeDoc Example

Welcome to the TypeDoc example! TypeDoc is a documentation generator for TypeScript.

TypeDoc automatically documents every variable, function, and class that is exported by your project. You can add explanations and examples to your documentation site by annotating your code with doc comments, e.g.

```
/**  
 * Calculates the square root of a number.  
 * @param x the number to calculate the root of.  
 * @returns the square root if `x` is non-negative or `NaN` if `x` is negative.  
 */  
export function sqrt(x: number): number {  
    return Math.sqrt(x);  
}
```

This project shows off some of TypeDoc's features:

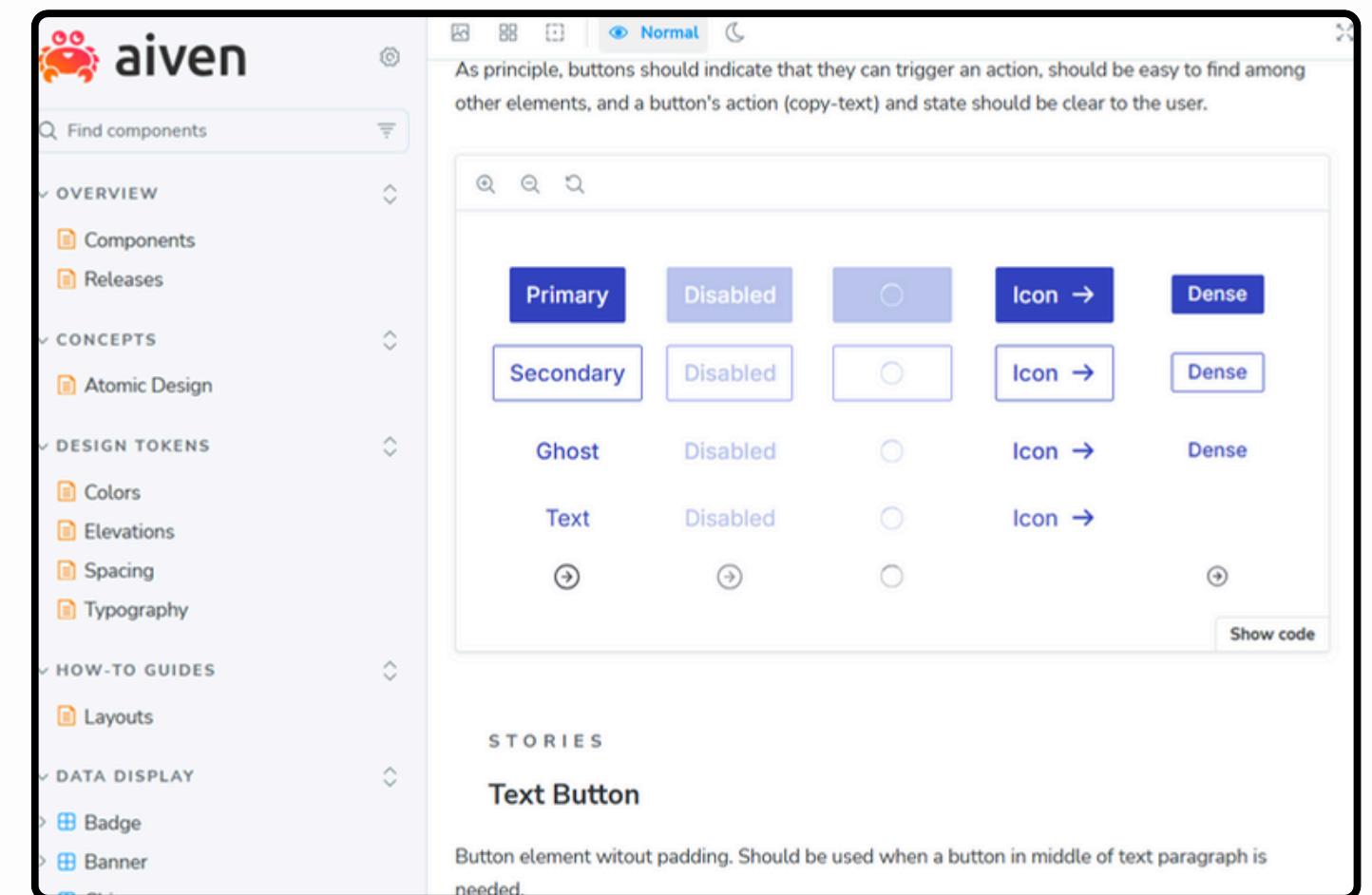


Visit link to learn more :  
[typedoc.org](https://typedoc.org)

# TOOLS

## STORYBOOK

- **UI components documentation**
- **Isolated environment for components**
- **Visual testing for props and states**
- **Collaboration between devs and designers**



Visit link to learn more :  
[storybook.js.org](https://storybook.js.org)

# BEST PRACTICES

- **Keep docs short and clear**
- **Always update with code changes**
- **Add practical code examples**
- **Treat documentation as code**

# ALAA SUFI

FrontEnd Developer

   @alaasufi00

"Code  
is written once,  
read hundreds of times."

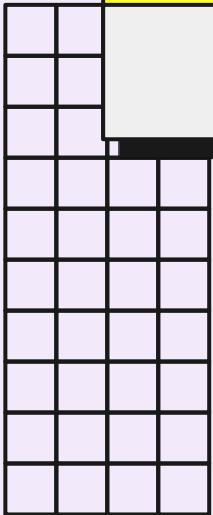


# **STREAMS IN JAVASCRIPT**

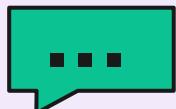
**PRESENTED BY:  
ENG. ABDULRHMAN ALSHAHIN**

Aleppo dev community

# WHAT IS A STREAM?



**Sequence of  
data/events  
over time**

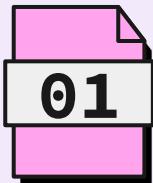


**YOU don't  
need to wait  
for all the  
data**

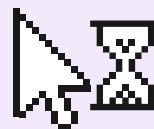


**Think of it  
as a  
'timeline'  
of events**

# Every day streams in js



Event  
listeners



Fetch api with  
readableStream



Node.js file  
streams



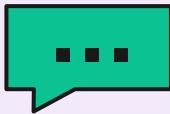
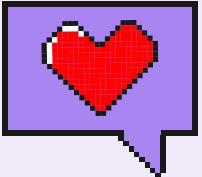
Web sockets  
(real-time messages)

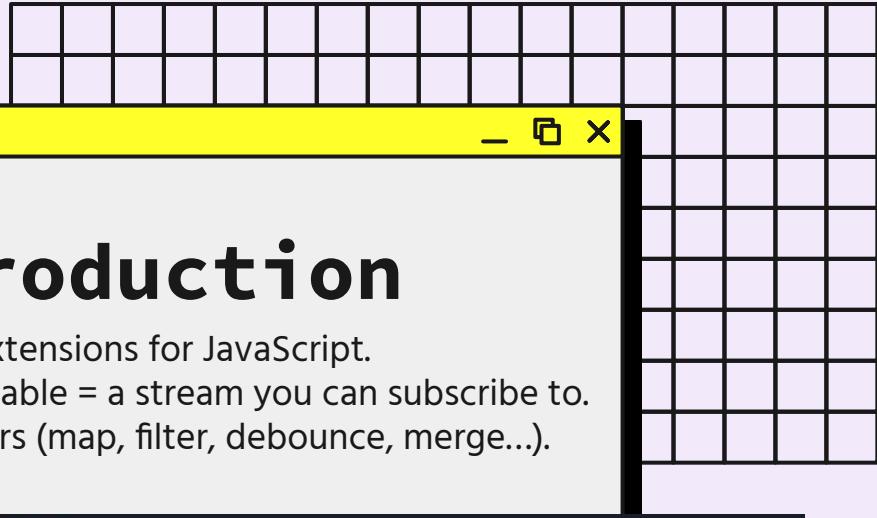
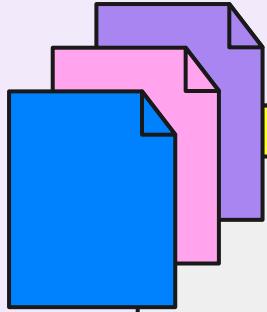
# Why Streams?

**Efficient  
memory usage**  
→ process  
chunks.

**Great for  
asynchronous  
& real-time  
data.**

**Live chats  
  
Music/video  
streaming**



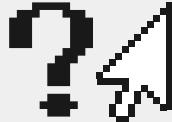


# RxJS Introduction

Reactive Extensions for JavaScript.

Core abstraction: Observable = a stream you can subscribe to.

Provides 100+ operators (map, filter, debounce, merge...).

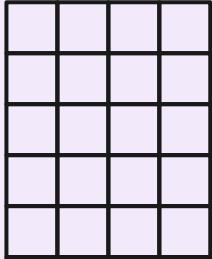


```
1  interval(1000)
2    .pipe(
3      take(5),
4      map(x => x * 2)
5    )
6    .subscribe(val => console.log(val));
7 // the result is: 0, 2, 4, 6, 8
```

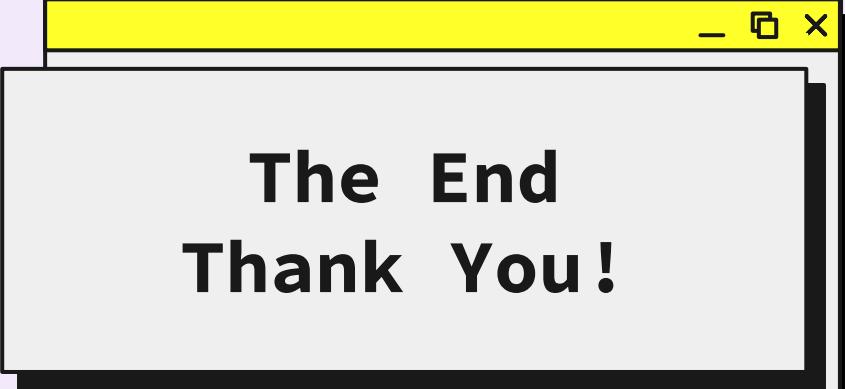


# RxJS Example

## School Students Data



```
1  fromEvent(schoolSelect, "change")
2    .pipe(
3      switchMap(school =>
4        fromEvent(branchSelect, "change").pipe(
5          switchMap(branch =>
6            from(fetch(`api/students?school=${school.target.value}&branch=${branch.target.value}`))
7              .then(res => res.json()))
8            )
9          )
10         )
11       )
12     .subscribe(data => console.log("Students:", data));
```



The End  
Thank You!

If you have any questions, feel free to  
reach out after the talk.

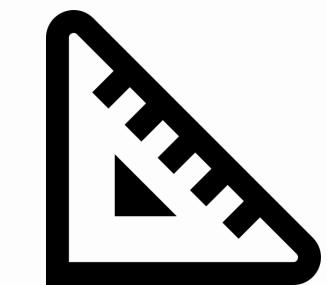
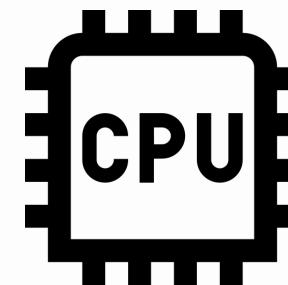
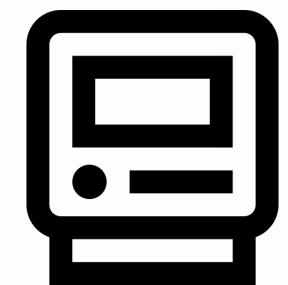
eng.abdulrhman.alshahin@gmail.com  
+963 940 481 087  
[linkedin.com/in/abdulrahman-alshahin](https://www.linkedin.com/in/abdulrahman-alshahin)





# CREATIVE CODING

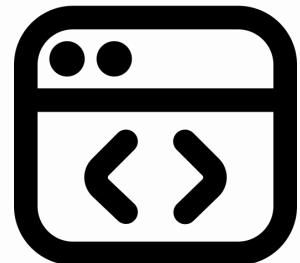
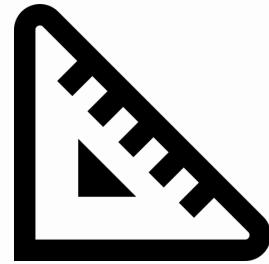
WHERE LOGIC MEETS ART



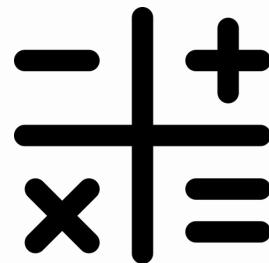
# WHAT IS CREATIVE CODING?

PROGRAMMING

$f_x$



01  
10

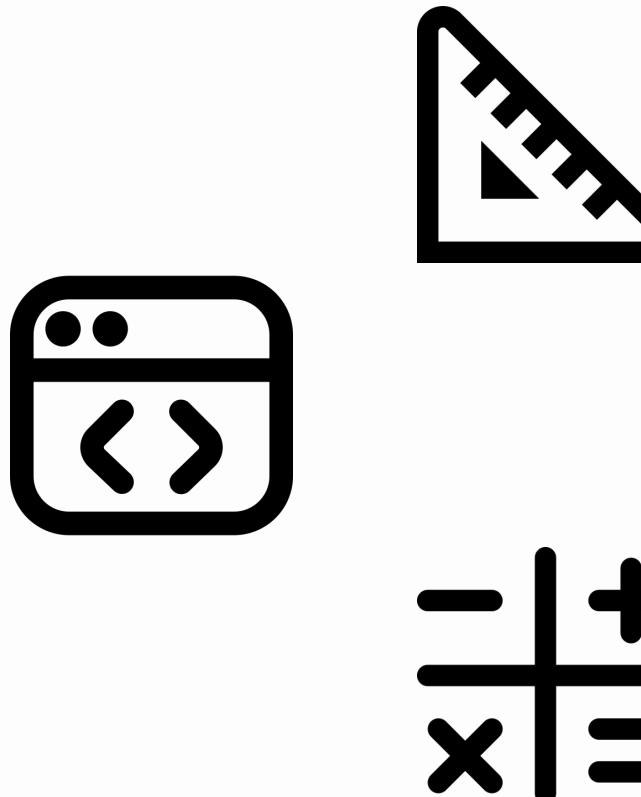


# WHAT IS CREATIVE CODING?

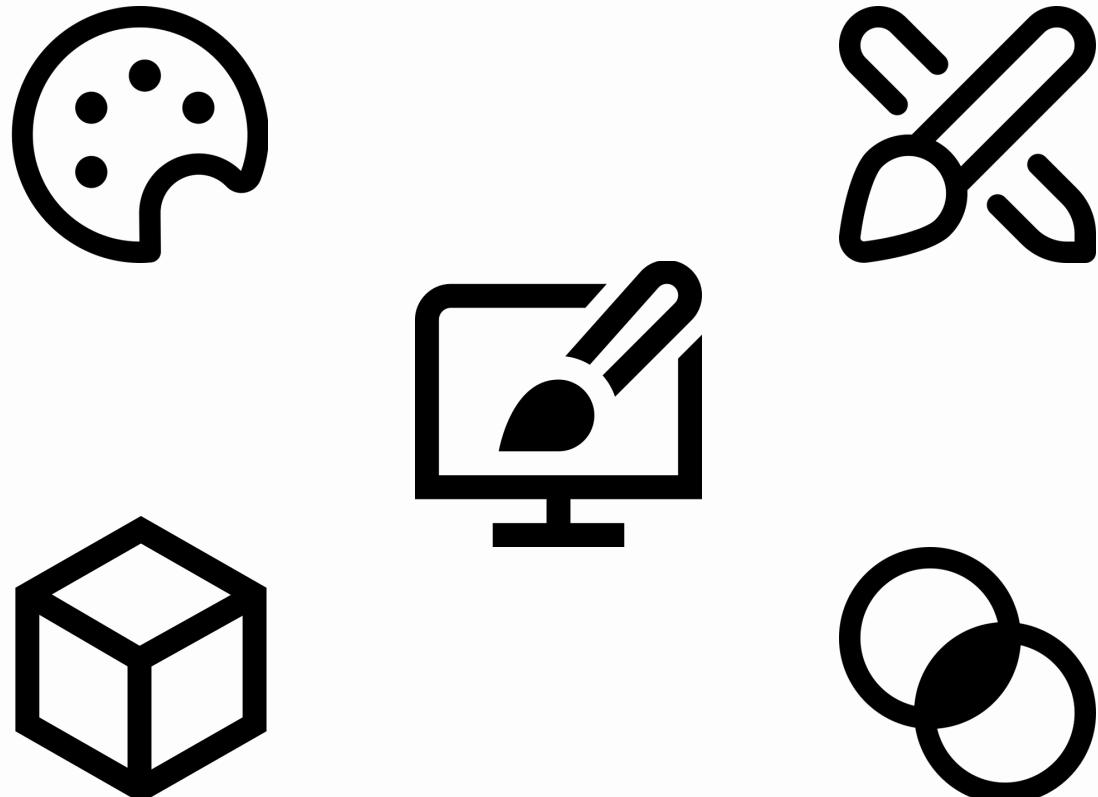
## PROGRAMMING

$f_x$

01  
10



## GRAPHIC DESIGN

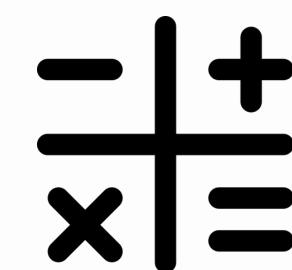
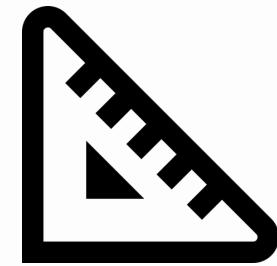
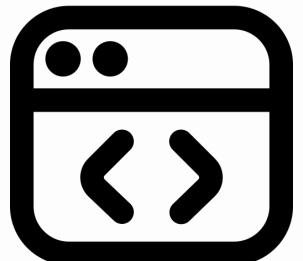


# CREATIVE CODING

## PROGRAMMING

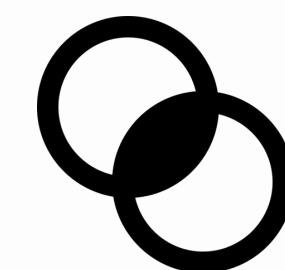
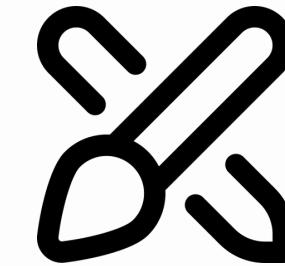
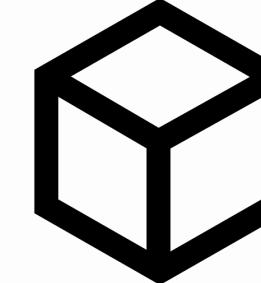
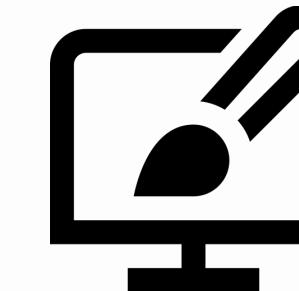
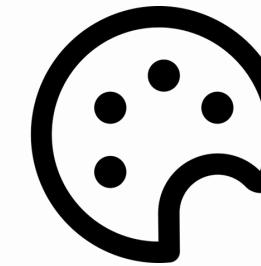
$f_x$

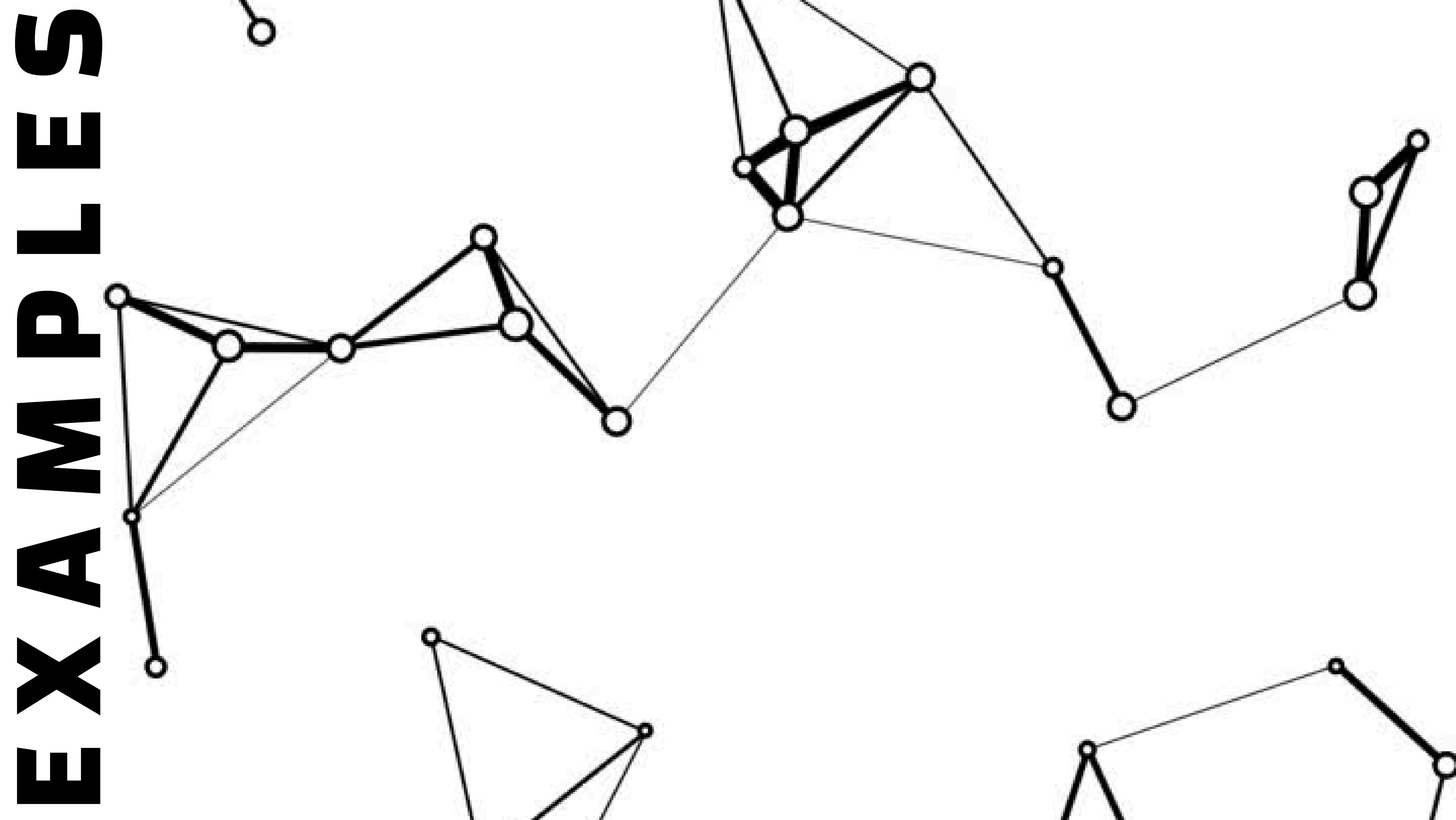
01  
10



+

## GRAPHIC DESIGN





EXAMPLES

W

W

L

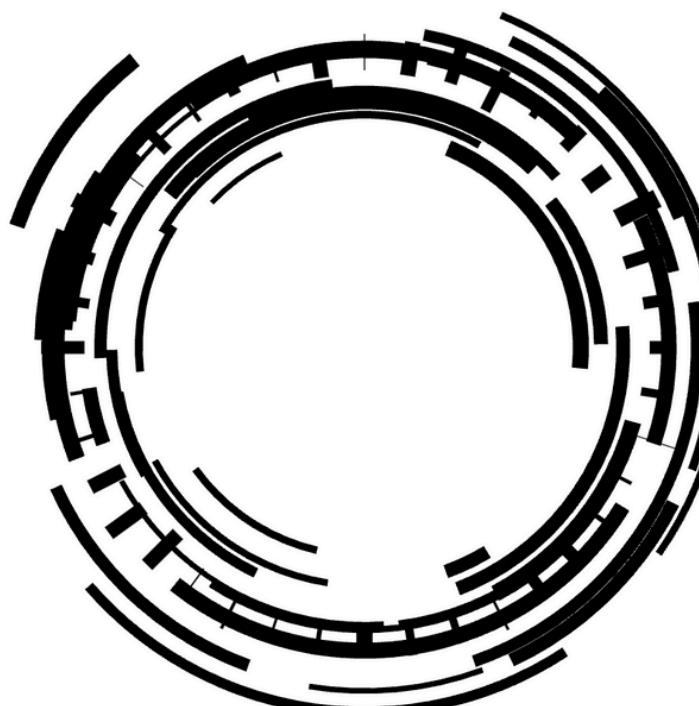
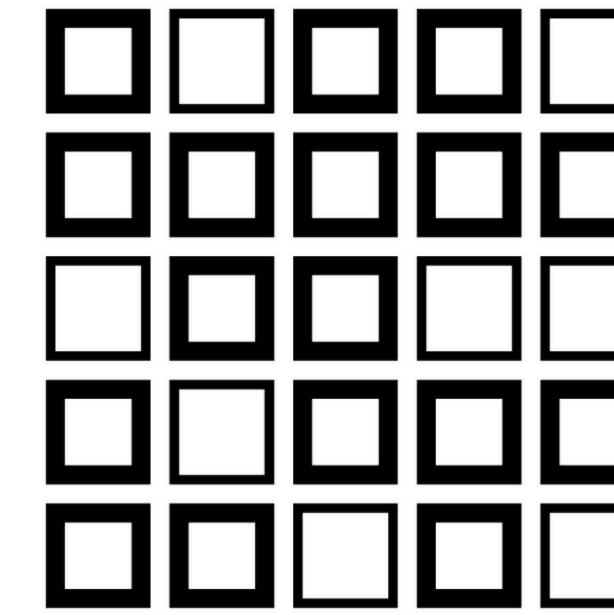
P

XAM

E

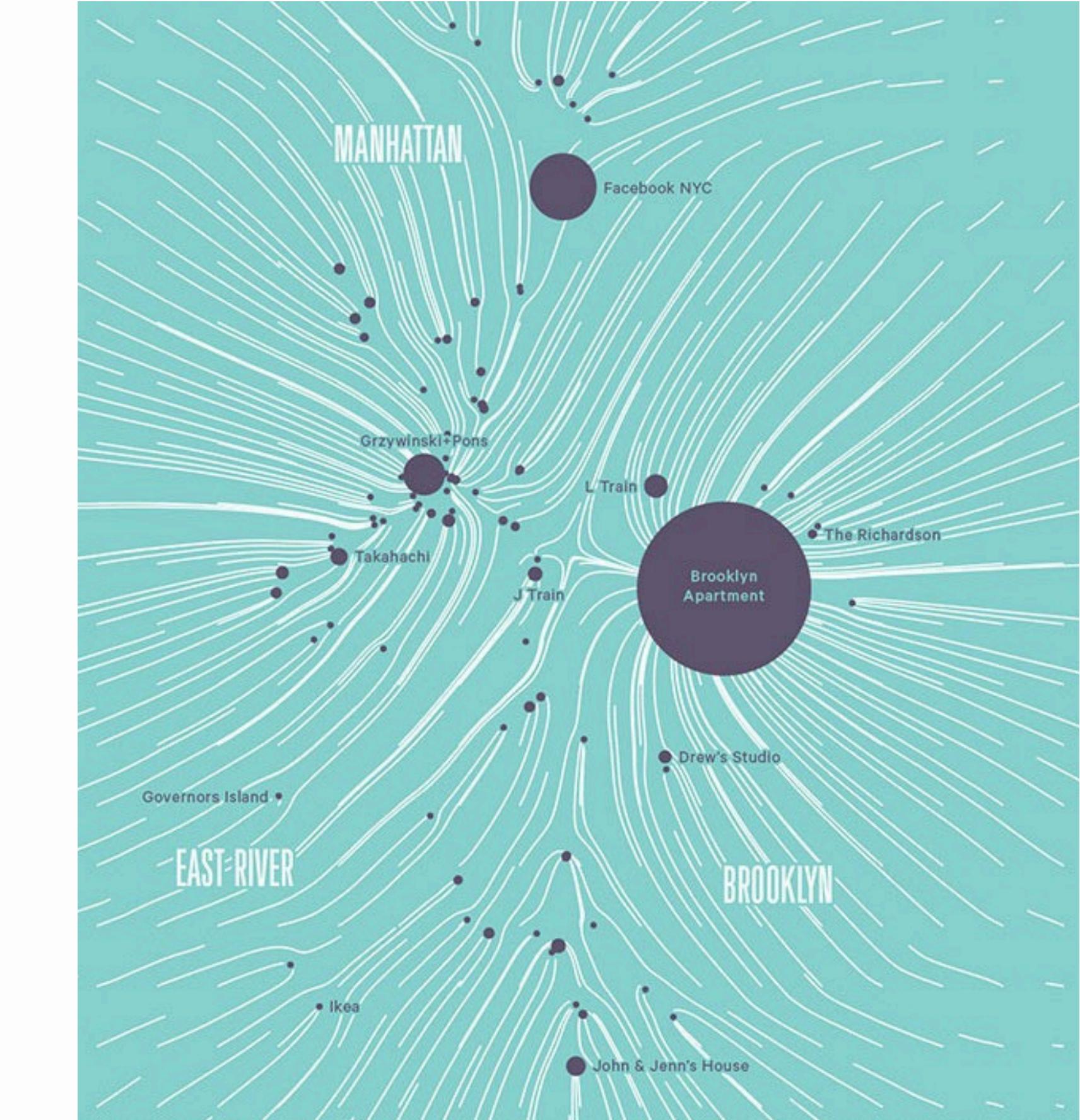
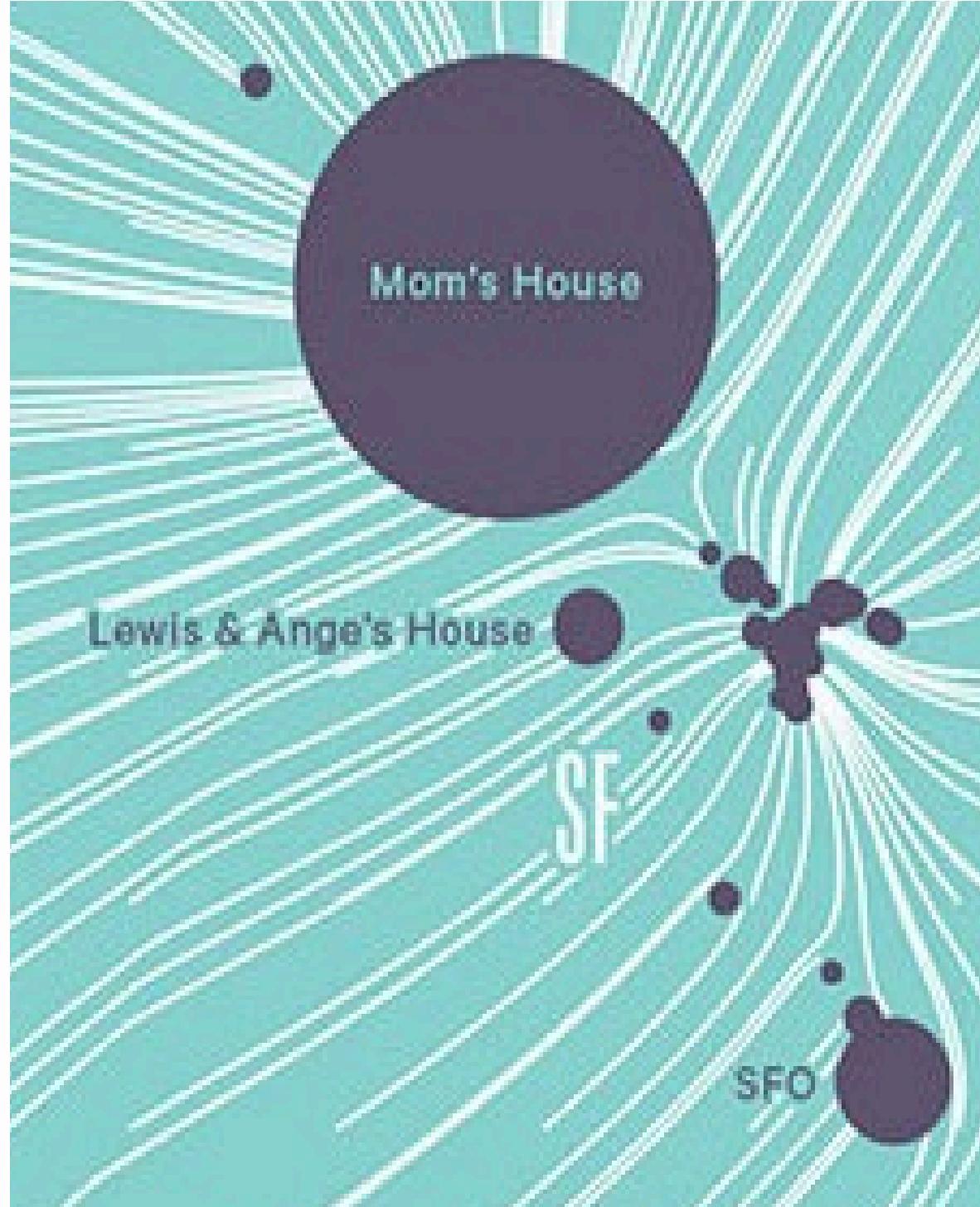
W

# GENERATIVE ART



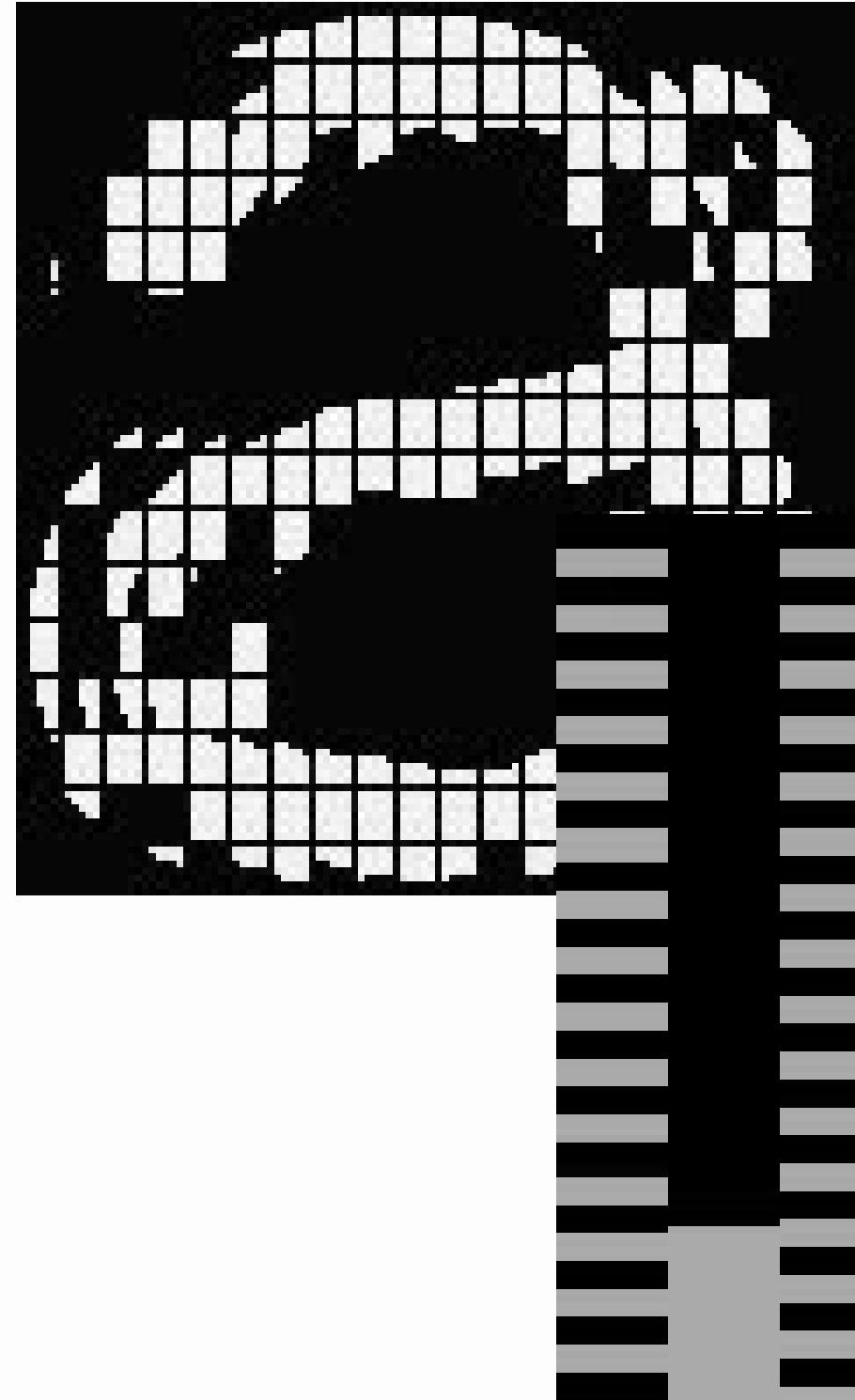
# DATA VISUALIZATION / DATA ART

## EXAMPLES



# **ANIMATIONS & MOTION GRAPHICS**

**EXAMPLE**



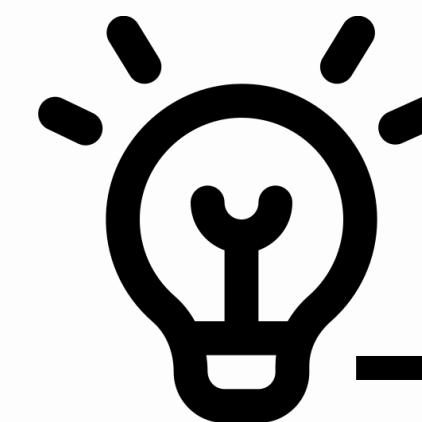
---

**CREATIVE  
CODERS**

---

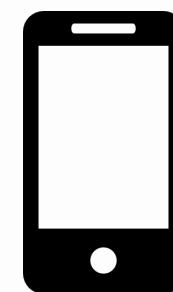
**UI/UX  
DESIGNERS**

**CUSTOMER**



DESIGNERS

**UI  
UX**



---

**CREATIVE  
CODERS**

---

**UI/UX  
DESIGNERS**

---

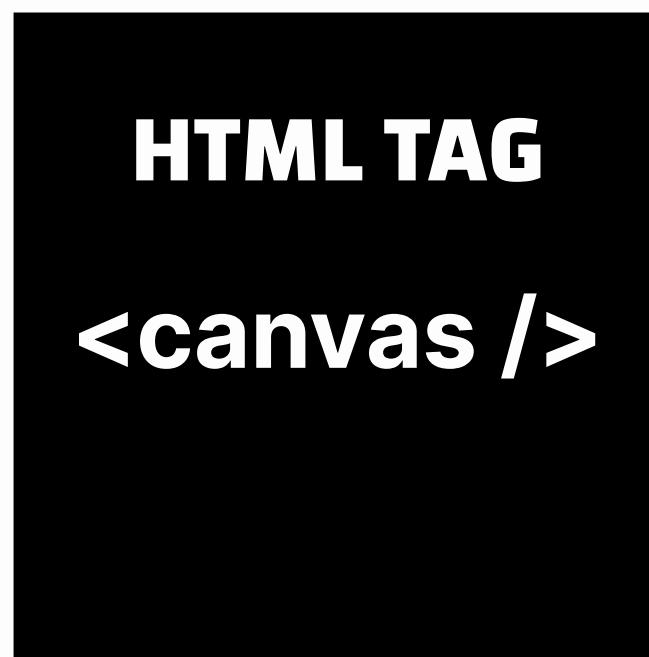


# TOOLS

## DRAWING FROM SCRATCH



p5.js



CANVAS - SKETCH

# TOOLS

## DRAWING FROM SCRATCH



p5.js



CANVAS - SKETCH

## READY-MADE TECHNIQUES



FRAMER MOTION



GSAP

# AMIR BATTAL

Software Engineering || Creative Developer



[amirbattal.com](http://amirbattal.com)



[amir\\_battal.js](https://www.instagram.com/amir_battal.js)



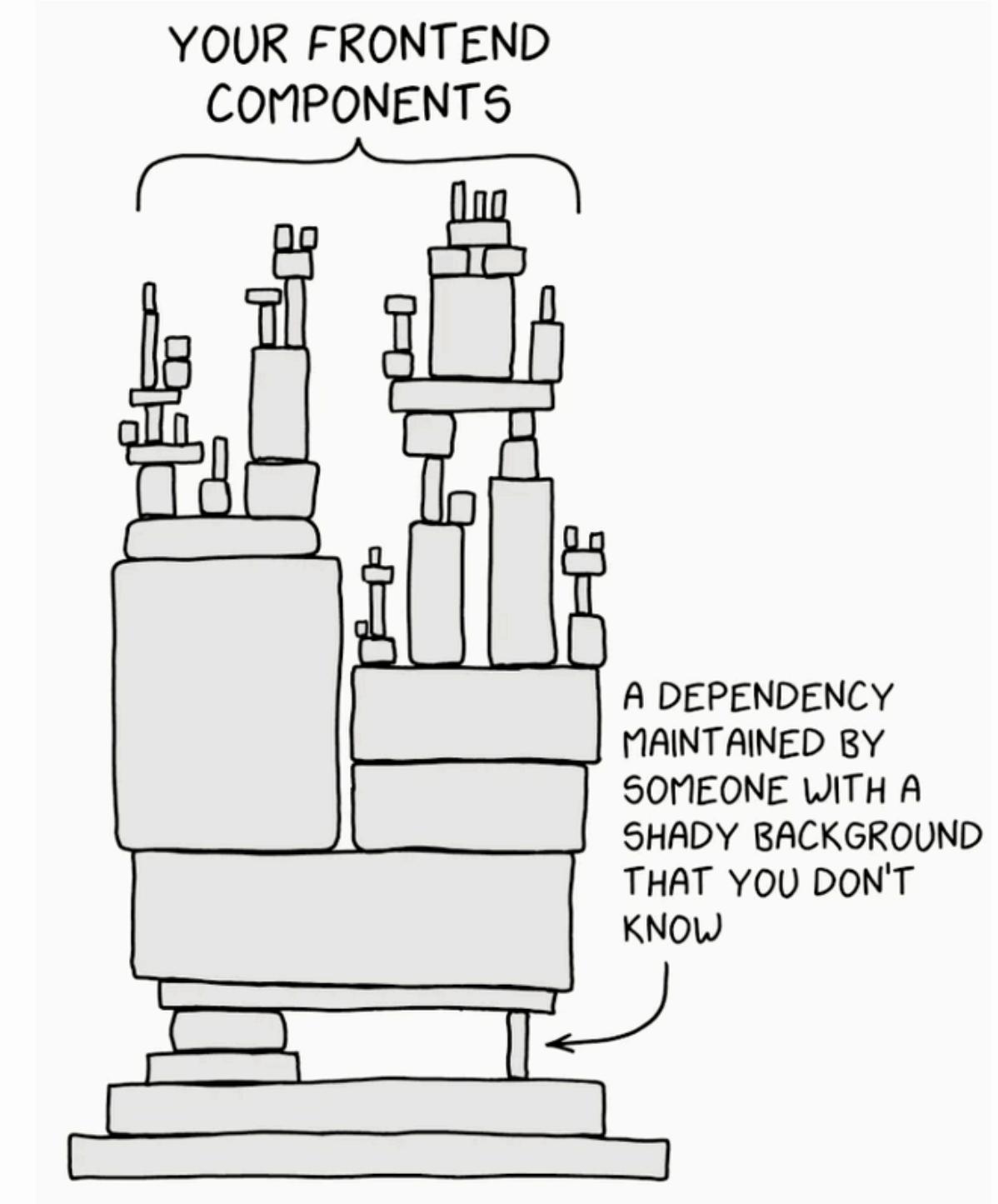
# THE RISK OF NPM PACKAGES

Is `npm install` dangerous?



Muzan Ghazy

Software Engineer



# KEY SECURITY ISSUES

- **Malicious Code Injection:** Nx, event-stream
- **Supply Chain Attacks:**
  - Maintainer Account Compromise: eslint-config-prettier
  - Dependency Confusion
- **Typosquatting/Fake Packages:** cross-env & crossenv
- **Abandoned or Unmaintained Packages :** lodash, left-pad
- **Over-Privileged Lifecycle Scripts**

# HOW TO MITIGATE NPM RISKS

- Run `npm audit`

```
@babel/runtime <7.26.10
Severity: moderate
Babel has inefficient RegExp complexity in generated code with .replace when transpiling named capturing groups - https://github.com/advisories/GHSA-968p-4vh-cqc8
fix available via `npm audit fix`
node_modules/@babel/runtime

axios 1.0.0 - 1.8.1
Severity: high
axios Requests Vulnerable To Possible SSRF and Credential Leakage via Absolute URL - https://github.com/advisories/GHSA-jr5f-v2jv-69x6
fix available via `npm audit fix`
node_modules/axios

next <=14.2.31
Severity: critical
Next.js authorization bypass vulnerability - https://github.com/advisories/GHSA-7gfc-8cq8-jh5f
Next.js Allows a Denial of Service (DoS) with Server Actions - https://github.com/advisories/GHSA-7m27-7ghc-44w9
Next.js Race Condition to Cache Poisoning - https://github.com/advisories/GHSA-qpjv-v59x-3qc4
Information exposure in Next.js dev server due to lack of origin verification - https://github.com/advisories/GHSA-3h52-269p-cp9r
fix available via `npm audit fix --force`
Will install next@14.2.32, which is outside the stated dependency range
node_modules/next

7 vulnerabilities (1 low, 2 moderate, 2 high, 2 critical)
```

# HOW TO MITIGATE NPM RISKS

- Run `npm audit fix` & Overrides

```
added 4 packages, removed 1 package, changed 14 packages, and audited 435 packages in 32s

143 packages are looking for funding
  run `npm fund` for details

# npm audit report

next <=14.2.31
Severity: critical
Next.js authorization bypass vulnerability - https://github.com/advisories/GHSA-7gfc-8cq8-jh5f
Next.js Allows a Denial of Service (DoS) with Server Actions - https://github.com/advisories/GHSA-7t
Next.js Race Condition to Cache Poisoning - https://github.com/advisories/GHSA-qpjv-v59x-3qc4
Information exposure in Next.js dev server due to lack of origin verification - https://github.com/
Authorization Bypass in Next.js Middleware - https://github.com/advisories/GHSA-f82v-jwr5-mffw
```

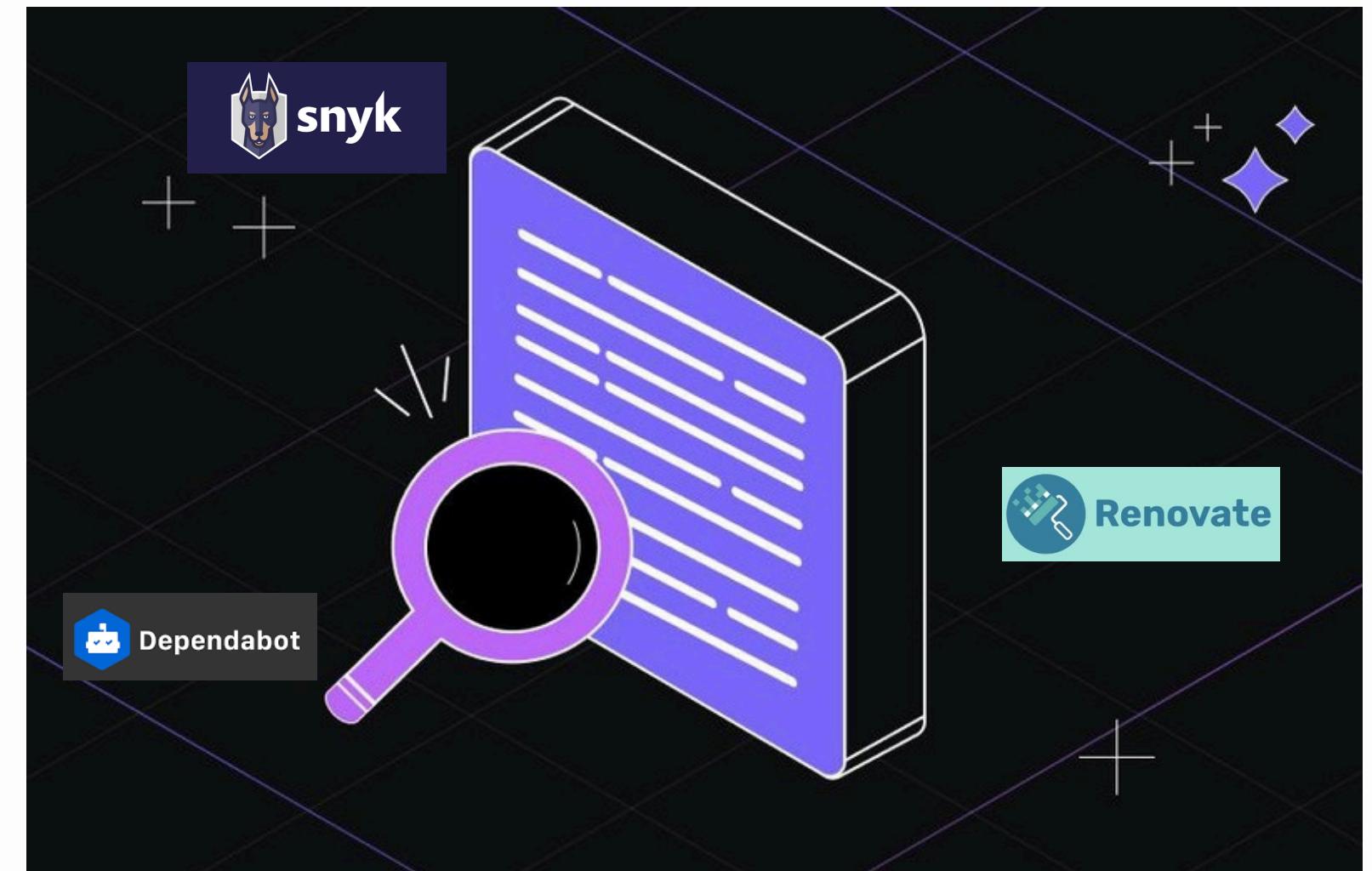
```
removed 7 packages, and audited 1633 packages in 28s
```

```
201 packages are looking for funding
  run `npm fund` for details
```

```
found 0 vulnerabilities
```

# HOW TO MITIGATE NPM RISKS

- Run `npm audit`
- Snyk
- Dependabot / Renovate
- Manual Version Locking
- Verify Package Names
- Minimize Unnecessary Dependencies



# OUTCOMES OF USING NPM SAFELY

- Fewer bugs & easier maintenance
- Safer user data
- Less risk from malicious or abandoned packages
- Faster, more stable frontend

**Frontend security is not optional—it's a daily practice** 





---

# Let's Build the Future of Aleppo's Tech Scene Together!

STAY CONNECTED!



---

aleppo.dev

