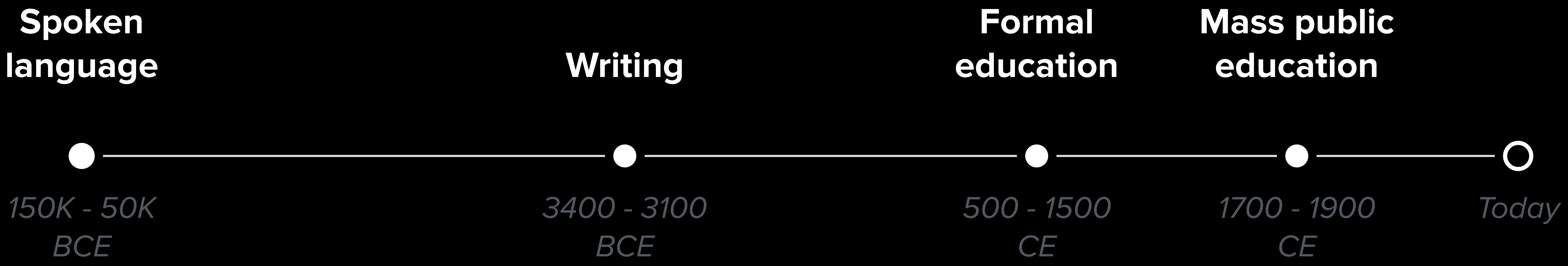
The Code Etymologist



Wikipedia (Origin of language, History of writing, History of education)

$$\begin{tikzcd}
\text{LATIN} & \text{LATIN} & \text{LATE ENGLISH} \\
\textbf{docere} \arrow[r] & \textbf{documentum} \arrow[r] & \textbf{document} \\
\textit{teach} & \textit{lesson, proof} &
\end{tikzcd}$$

Oxford Languages via. Google search

Today

ANDREI PFEIFFER

Code Etymologist

There is no documentation!

- Software Developers

There is no documentation!

Why is that?

Because nobody wrote it.

Agile Development

The Manifesto for Agile Development

1. ***Individuals and interactions*** *over processes and tools*
2. ***Working software*** *over comprehensive documentation*

agilemanifesto.org

The Manifesto for Agile Development

6th principle

The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.

agilemanifesto.org /principles.html

The Manifesto for
Agile Development

We embrace documentation, but not hundreds of pages of never-maintained and rarely-used tomes.

agilemanifesto.org /history.html

- Project setup
- Coding guidelines
- Development workflows
- Features
- Data structures
- UI Components library
- Technical decisions

DISCLAIMER

For internal documentation purposes only

- Project setup
- Coding guidelines
- Development workflows
- Features
- Data structures
- UI Components library
- Technical decisions

Project setup

Basic setup

For npm-based projects

```
git clone https://...
npm install
npm start
```

README.md

```
...
## Local development
git clone https://...
npm install
npm start
```

Additional steps

- Prerequisites
- Database server(s)
- Environment variables

README.md

Missing Node.js version

```
...
## Prerequisites
- Node.js (https://nodejs.org/en/download)
```

Me (private message)
Hey, what Node version do you use?

AA
Anya Alston
20

Me (group message)
Hey, I’m curious what Node version do you use for this project?

D C BB
Team mates
22, 18, 21

EE
Emery Ellis
Latest & greatest

FF
Felix Ford (private message)
Don't tell the others, I use Bun 🤫

Me (group message)
Hey, I’m curious what Node version do you use for this project?

Team mates
22, 18, 21

Emery Ellis
Latest & greatest

Felix Ford (private message)
Don't tell the others, I use Bun 🤫

DevOps / Infrastructure team
I see you’re using Node. What version should we update to?

Team members
🤔 ...

README.md

Documented Node.js version

```
...
## Prerequisites
- Node.js (v22.14.x)
```

Node.js version

Enforced

```
// package.json
{
  "engines": {
    "node": "22.14.x"
  }
}
```

```
// .npmrc
engine-strict=true
```

Node.js version

Mismatch error

```
npm ERR! engine Unsupported engine
npm ERR! notsup Not compatible with your version of node/npm
npm ERR! notsup Required: {"node":"22.14.x"}
npm ERR! notsup Actual:   {"npm":"9.8.1","node":"18.18.2"}
```

README.md

Referenced Node.js version

```
...
## Prerequisites
- Node.js (check package.json > engines for version)
```

README.md

```
...
## Prerequisites
- Node.js (check package.json > engines for version)
- MariaDB (v11.4.6)
- Redis (v7.4.6)
```

Dockerfiles

```
version: "3.8"

services:
  mysql:
    image: mariadb:11.4.6
    ports:
      - 55008:3306
    environment:
      DB_PASS: idiguplostknowledge
      DB_NAME: local-db

  redis:
    image: redis:7.4.6
    ports:
      - 55006:6379
```

README.md

```
...
## Prerequisites
- Node.js (check package.json > engines for version)
- Docker (https://www.docker.com/)

## Local development
docker compose up -d
...
```

.env

```
SOME_API_KEY = "437a3935-f1ea-4348-b6a8-9e7f43910ff1"
SOME_SERVICE_URL = "https://dev.eu.smth.com"
ENABLE_METRICS = "true"
```

server.js

```
import 'dotenv/config';
```

.env.sample

```
SOME_API_KEY = ""
SOME_SERVICE_URL = ""
ENABLE_METRICS = "false"
```

env.js

```
import { z } from 'zod';

export const envSchema = z.object({
  SOME_API_KEY: z.string(),
  SOME_SERVICE_URL: z.string(),
  ENABLE_METRICS: z.boolean().optional().default(false),
});
```

server.js

```
import 'dotenv/config';
import { envSchema } './env';

envSchema.parse(process.env);
```

Typesafe environment variables with Zod

Jacob Paris

https://www.jacobparis.com/content/type-safe-env

👍

Write

🤘

Code

Data Structures

Values vs. Types

Values	Types
Dynamic analisys	Static analisys
Unit tests	Type checkers
Runtime	Compile time
Reality	*Theory*
*"How things **actually are**"*	*"How things **should be**"*

```
async function getMessages() {
  const res = await fetch(`/api/messages`);
  return res.json();
}
```

```
async function getMessages() {
  const res = await fetch(`/api/messages`);
  return res.json();
}

// runtime response
[
  {
    "id": 2025,
    "content": "Good morning, JSHeroes!",
1️⃣  "status": 2,
2️⃣  "messageType": "text",
3️⃣  "annotation": null
  }
]
```

```
async function getMessages(): Array<Message> {
  const res = await fetch(`/api/messages`);
  return res.json();
}
```

```
async function getMessages(): Array<Message> {
  const res = await fetch(`/api/messages`);
  return res.json();
}

type Message = {
  id: number;
  content: string;
1️⃣  status: number;
2️⃣  messageType: string;
3️⃣  annotation: null | Annotation;
};
```

1️⃣ Enums

```
type Message = {
  id: number;
  content: string;
  status: number;
  messageType: string;
  annotation: null | Annotation;
};
```

```
enum MessageStatus {
    Unsent = 0,
    Sent = 1,
    Deleted = 2,
};
```

1️⃣ Enums

```
type Message = {
  id: number;
  content: string;
✅ status: MessageStatus;
  messageType: string;
  annotation: null | Annotation;
};
```

2️⃣ Literal Unions

```
type Message = {
  id: number;
  content: string;
  status: MessageStatus;
✅ messageType: "text" | "comment";
  annotation: null | Annotation;
};
```

3️⃣ Discriminated Unions

```
type Message = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "text" | "comment";
  annotation: null | Annotation;
};
```

3️⃣ Discriminated Unions

```
type Message = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "text" | "comment";
  annotation: null | Annotation;
};
```

3️⃣ Discriminated Unions

```
type Message = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "text";
  annotation: null;
};
```

```
type Message = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "comment";
  annotation: Annotation;
};
```

3️⃣ Discriminated Unions

```
type TextMessage = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "text";
  annotation: null;
};
```

```
type CommentMessage = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "comment";
  annotation: Annotation;
};
```

✅ `type Message = TextMessage | CommentMessage;`

```typescript
enum MessageStatus { Unsent = 0, Sent = 1, Deleted = 2 };

type TextMessage = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "text";
  annotation: null;
};

type CommentMessage = {
  id: number;
  content: string;
  status: MessageStatus;
  messageType: "comment";
  annotation: Annotation;
};

type Message = TextMessage | CommentMessage;
```

Values vs. **Types**

Values and **Types**

Schemas

```
enum MessageStatus { Unsent = 0, Sent = 1, Deleted = 2 };

const TextMessage = z.object({
  id: z.number(),
  content: z.string(),
  status: z.enum(MessageStatus),
  messageType: z.literal("text"),
  annotation: z.null(),
});

const CommentMessage = z.object({
  id: z.number(),
  content: z.string(),
  status: z.enum(MessageStatus),
  messageType: z.literal("comment"),
  annotation: Annotation,
});

const Message = z.discriminatedUnion("messageType", [TextMessage, CommentMessage]);
type Message = z.infer<typeof Message>;
```

Extra code is fine,
as long as it communicates
meaningful information

Andrei Pfeiffer (May 2025, JSHeroes)

✅ Project setup

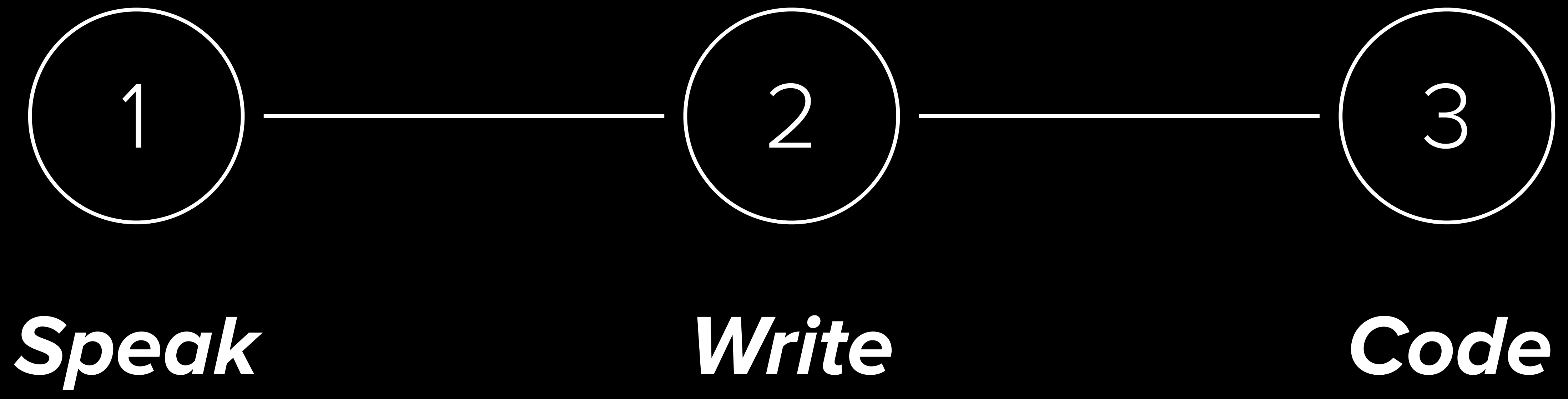
- Coding guidelines
- Development workflows
- Features

✅ Data structures

- UI Components library
- Technical decisions

The code etymologist

andreipfeiffer.dev/*blog*


1
Speak
2
Write
3
Code

1 2 3

Speak ***Write*** ***Code***

We all blame the lack of documentation.
We all appreciate good documentation.

- Software Developers

THANK YOU

andreipfeiffer.dev