1.2.2. Jak tworzyć dokumentację?

Ułatwianie korzystania z technologii sobie i innym

Instrukcje

- Przepis na…
- Checklist’y np. https://frontendchecklist.io/
- Brak wyjaśnień "dlaczego"

Komentarze w kodzie

- JSDoc (https://devdocs.io/)
- Dodatkowe wyjaśnienia

Przykład JSDoc

```
/**
 * @typedef {Object} User
 * @property {number} id
 * @property {string} email
 * @property {?string} firstName
 */

/**
 * Gets user from the database
 * @param {string} email
 * @returns {?User}
 */
function getUser(email) {
  try {
    const user = getUserFromDB(email)
    return user
  } catch (error) {
    return null
  }
}
```

Przykład TypeScript

```typescript
interface User {
  id: number
  email: string
  firstName?: string
}

function getUser(email: string): User | null {
  try {
    const user = getUserFromDB(email)
    return user
  } catch (error) {
    return null
  }
}
```

Generowanie dokumentacji z JSDoc

```
npx jsdoc
```

Pliki README

- https://github.com/axios/axios

Wiki

- https://github.com/ohmyzsh/ohmyzsh/wiki
- https://www.notion.so/MIKR-US-Don-t-Panic-5c3bdde2e0b545e7866524fc117446c3
- https://wiki.htw-berlin.de/confluence/display/hilfe/Vorlagen
- https://www.mediawiki.org/wiki/MediaWiki

Generatory

- https://swagger.io/ (Tpay)
- https://docusaurus.io/ (Playwright, Jest)
- https://squidfunk.github.io/mkdocs-material/ (AWS Copilot)
- https://www.sphinx-doc.org/en/master/ (CodeIgniter)
- https://alternativeto.net/software/sphinx/

Testy

- Podnoszą jakość wyłapując błędy
- Dokumentują działanie kodu poprzez przykłady użycia