

## 1.7.2. **Dobre praktyki**

Wyższa jakość pracy Web Developer'a

# Dostępność (a11y)

Umożliwienie swobodnego korzystania z aplikacji przez wszystkich

- <https://developer.mozilla.org/en-US/docs/Web/Accessibility>
- <https://webaim.org/>
- <https://www.w3.org/TR/WCAG2/>

# Internacjonalizacja, lokalizacja (i18n)

Wybór języka, waluty, prezentacja daty i godziny, strefy czasowe

- <https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Internationalization>
- [https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/Date](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Date)
- <http://i18njs.com/>
- <https://www.i18next.com/>

# Enkapsulacja

Hermetyzacja, izolacja

- Chowanie technicznych rozwiązań za interfejsem
  - Pilot do TV
  - Prowadzenie samochodu
  - API
- Jedno miejsce - jak coś zrobić (programowanie imperatywne)
  - Zewnętrzne biblioteki
- Wiele miejsc może wykonywać polecenia - co zrobić (programowanie deklaratywne)
  - Wykorzystywanie API
- Dużo programowania deklaratywnego mało imperatywnego
- Jeśli coś jest do wszystkiego to jest do niczego
  - Pojazd: taksówka, czołg, walec, bolid wyścigowy, koparka, motocykl

# Wzorce projektowe

Gotowe rozwiązania problemów

- [https://en.wikipedia.org/wiki/Software\\_design\\_pattern](https://en.wikipedia.org/wiki/Software_design_pattern)
- <https://refactoring.guru/design-patterns/catalog>

# Wersjonowanie

Cykle wydań i changelog

- major.minor.patch
- [https://code.visualstudio.com/updates/v1\\_71](https://code.visualstudio.com/updates/v1_71)
- <https://reactjs.org/versions/>
- <https://ubuntu.com/about/release-cycle>

# Automatyzacja

Kiedy warto automatyzować, a kiedy nie?

- Lintery, formatery (ESLint, Prettier)
- Skrypty (npm scripts, Node.js CLI, bash)
- Testy automatyczne
- <https://xkcd.com/1205/>