

Multiplatform szoftverfejlesztés

Összehasonlítás

React vs Vue vs Angular

Tudás

- Angular teljes keretrendszer
 - Nehezebb megtanulni
- React és Vue csak UI
 - Rátehetők az alkalmazás egy részére is
 - Gyorsan el lehet kezdeni használni őket
 - Van sok (főleg React-hoz) könyvtár
 - Ezekkel fel tudjuk hízlalni őket Angular szintre
 - Nem triviális mindent összehangolni

Sebesség

- Szinte azonos – ez alapján nem lehet dönteni
- Talán Vue a leggyorsabb (nagyon kicsi eltérés)
 - Főleg, ha nem figyelünk React-ben (PureComponent)
- Preact nagyon hasonló React-hez
 - Gyorsabb mindegyiknél
 - 9KB (3KB tömörítve)
 - Csak az alapokat tudja
- Nulláról megírva a kódot tudunk még gyorsabbat írni...

HTML

- Vue és Angular HTML sablonnal dolgozik
 - Deklaratív felület megadás
 - Bár tudunk kódot írni (kifejezést), de csak bizonyos helyekre
 - Attribútumokkal szabályoz
 - Feltételes elemek és ciklusok
- React nem tisztán deklaratív
 - Kódot írhatunk a felület leíró logikába (JSX, TSX)
 - JS/TS kód szabályozza a teljes rendszert
 - Ez elvileg többet tud
 - A valóságban ritka, amikor e nélkül nem megy

CSS

- Vue támogat CSS-t a .vue fájlban
 - Lehet scoped (csak arra a komponensre hat)
 - Nem feltétlen hasznos, attól függ, hogy ki csinálja a CSS-t
 - Ha designer, akkor nem hasznos
 - Ha a fejlesztő, akkor igen
- Angular komponensenkénti CSS-t használ
- React-ban alapban nincs ilyen
 - De vannak külső könyvtárak CSS-in-JS
 - Pl. emotion
 - Ezeknek általában van futásidejű költsége
 - Lehet fordítás idejű a CSS feldolgozás: Webpack, stb.

Fordítás

- Mindegyiket fordítani kell
 - React: JSX/TSX
 - Vue: .vue fájl
 - Angular: rengeteg fájl csomagolni
- Elvileg lehet olyan kódot írni, amihez nem kell fordító
 - Általában borzalmas fejlesztői élmény
- Nem nagy gond a fordítás
 - Amennyire nehéz beállítani az egész rendszert

Nyelv

- JavaScript
 - Angular nem használható
 - Mindegyik másik igen
- TypeScript
 - Angular TS-ben van írva, és ezt támogatja
 - Vue (3.0) TS-ben van írva, mindent támogat
 - React JS-ben van írva, mindent támogat
 - Még TSX is van hozzá
 - Facebook Flow-t használ, ami típusos JS (nem igazán terjedt el)

Méret

- Angular a legnagyobb
 - Tree-shaking után is
- Vue és React hasonló, kisebb
 - De ha hozzáveszünk sok könyvtárat, hogy Angular képességűek legyenek, akkor megnőnek ezek is
- Preact és hasonlók nagyon kicsik

Csomagolás

- Mindegyik tudja
 - Lazy loading
 - Code splitting
 - Tree shaking
 - Minimalizálás
 - Stb.
- Webpack, Parcel, Rollup
 - Alapban a project létrehozó eszköz Webpack-et állít be
- ES csomagolóknak kezelik Vue-t és React-et

Komplex állapot kezelés

- Mindegyikhez van
 - Vue: Vuex
 - React: Redux
 - Angular: alapban benne van
- Nem feltétlen kell
 - Ha a komponenseknek csak a saját állapotuk kell
 - Vagy esetleg egy globális állapot elég
 - Nagyobb alkalmazásnál valami kell
 - Lehet saját megoldás is

IDE

- Sok szerkesztő van, általában amelyik tudja az egyiket, az tudja a másikat is
- Közepes eszközök
 - Ha valaki a régi IDE-ből jön, akkor jók
 - C# fejlesztőnek gyengék
 - Debuggolással gond van
 - Nem ott áll meg, nem jól lép át, nem jó a hibaüzenet
 - Csomagolás és konfiguráció problémás
 - Hozzáértést igényel
 - Nem feltétlen gyors
 - CLI megoldások: rugalmas, de bonyolult

Kérdések?